IMAE FOR NOISE-ROBUST LEARNING: MEAN ABSOLUTE ERROR DOES NOT TREAT EXAMPLES EQUALLY AND GRADIENT MAGNITUDE'S VARIANCE MATTERS

Xinshao Wang[*†]
University of Oxford
xinshaowang@gmail.com

Yang Hua
Queen's University Belfast
y.hua@qub.ac.uk

Elyor Kodirov

ekodirov@gmail.com

David A. Clifton[‡]
University of Oxford
davidc@robots.ox.ac.uk

Neil M. Robertson
Queen's University Belfast
n.robertson@qub.ac.uk

ABSTRACT

In this work, we study robust deep learning against abnormal training data from the perspective of example weighting built in empirical loss functions, i.e., gradient magnitude with respect to logits, an angle that is not thoroughly studied so far. Consequently, we have two key findings: (1) Mean Absolute Error (MAE) Does Not Treat Examples Equally. We present new observations and insightful analysis about MAE, which is theoretically proved to be noise-robust. First, we reveal its underfitting problem in practice. Second, we analyse that MAE's noise-robustness is from emphasising on uncertain examples instead of treating training samples equally, as claimed in prior work. (2) The Variance of Gradient Magnitude Matters. We propose an effective and simple solution to enhance MAE's fitting ability while preserving its noise-robustness. Without changing MAE's overall weighting scheme, i.e., what examples get higher weights, we simply change its weighting variance non-linearly so that the impact ratio between two examples are adjusted. Our solution is termed Improved MAE (IMAE). We prove IMAE's effectiveness using extensive experiments: image classification under clean labels, synthetic label noise, and real-world unknown noise.

1 INTRODUCTION

In this work, we target at robust deep learning, which is indispensable when it comes to large-scale industrial applications. It is non-affordable to guarantee the quality of training data as its scale grows dramatically. Consequently, abnormal examples[1] generally exist in large-scale real-world scenarios Berrada et al. (2018), which is caused by many factors, such as incomplete annotation, wrong labelling, subjectiveness, bias and so forth. Unfortunately, DNNs trained with categorical cross entropy (CCE) can fit random patterns Zhang et al. (2017).

[*]This work was mainly done at Queen's University Belfast and University of Oxford.

[†]For the source code, based on the requests for academic research and kindness to cite our work, we will release and maintain it in https://github.com/XinshaoAmosWang/DeepCriticalLearning.

[‡]Prof. David A. Clifton was supported by the NIHR Oxford Biomedical Research Centre, the InnoHK Hong Kong Centre for Cerebro-cardiovascular Health Engineering (COCHE), and the Pandemic Sciences Institute at the University of Oxford. Prof. David A. Clifton was also funded by an NIHR Research Professorship and an RAEng Research Chair.

[1]A training example is denoted as an observation-label pair, where the observation can be an image or video while the label defines its semantic information. We regard a training example as abnormal unrestrictedly whenever its observation and label are semantically unmatched, e.g., out-of-distribution examples (the observations contain only background or objects that do not belong to any training class), or examples with wrongly annotated labels.

Table 1: Classification accuracy (%) of CCE, MAE, and IMAE on CIFAR-10 Krizhevsky (2009). 40% of training examples, i.e., the noisy subset, have wrong labels. We test each model's performance on test set, noisy subset and clean subset of training data. The backbone is ResNet56 owning enough capacity He et al. (2016).

Loss	Test set (Generalisation)	Noisy subset (Noise-tolerance)	Clean subset (Learning ability)
CCE	63.3	75.0	96.2
MAE	66.9	8.1	**74.3** (worst)
IMAE	**81.5** (best)	**6.5** (best)	93.1

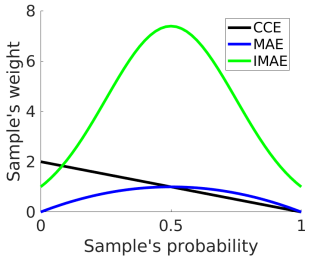


Figure 1: Sample's weight along with sample's probability being classified to its labelled class in CCE, MAE, IMAE with $T = 8$. If probabilities are uniformly distributed, the variances of CCE's, MAE's and IMAE's weighting curves are 0.33, 0.09 and 4.55, respectively.

Great advances have been made towards training DNNs robustly when abnormal training examples exist Arpit et al. (2017); Chang et al. (2017); Ren et al. (2018); Jiang et al. (2018). The robust loss function is one of them. In this paper, we study a so-claimed robust loss function, mean absolute error (MAE) following Ghosh et al. (2017); Zhang & Sabuncu (2018). According to the theoretical analysis of CCE and MAE in Ghosh et al. (2017), CCE is sensitive to label noise while MAE is noise-tolerant. Thereafter, generalised cross entropy (GCE) Zhang & Sabuncu (2018) concludes MAE treats training samples equally, thus being noise-robust.

However, our empirical observation and technical analysis lead us to a contradictory and more reasonable conclusion.
Observation: In Table 1, when 40% noise exists, compared with CCE, MAE underfits to clean training data points, thus fitting much fewer abnormal examples.
Conclusion: In Figure 1, MAE emphasises more on uncertain examples, whose probabilities of being classified to its labelled class are around 0.5, thus being noise-robust.

Specifically, according to Table 1, MAE is much more noise-tolerant than CCE. However, its ability of learning meaningful patterns is much weaker, fitting only 74.3% of the clean subset. We provide an intuitive interpretation for this according to Figure 1: *The variance of MAE's weight curve along with probability is only 0.09. As a result, the impact ratio between two examples is too small.*[2] The impact ratio reflects the relative impact of one example versus another for updating parameters. Due to MAE's small weight variance, informative samples cannot contribute enough against non-informative ones. Therefore, MAE cannot learn meaningful patterns well and is not widely used.

To adjust MAE's weight variance, we design an effective and simple solution, IMAE, which non-linearly transforms MAE's weighting scheme by an exponential function. On the one hand, by preserving MAE's overall weighting scheme, IMAE is noise-robust. On the other hand, by making the gradient magnitude's variance over training examples controllable, it learns meaningful patterns much better.

We demonstrate the effectiveness of IMAE under different scenarios. Most importantly, these empirical evidences justify that our interpretation of MAE's underfitting problem is reasonable and our proposed solution is superior. Our key findings are summarised as follows:

- CCE overfits to noise easily because it emphasises on low-probability examples to which abnormal ones generally belong. Although CCE's weight variance is not large (0.33), its fitting ability benefits from emphasising on low-probability examples.
- MAE is noise-robust by focusing on uncertain (medium-probability) examples instead of treating all equally. However, MAE generally underfits due to its small weights variance (0.09), leading to small impact ratio between even far different examples.
- Our proposed IMAE achieves new state-of-the-art on robust training against synthetic label noise and realistic unknown noise simply by adjusting MAE's weight variance, which is inspiring.

[2]The terms, examples' weight or impact, and examples' gradient magnitude w.r.t. logits, are used interchangeably because we define the weight by gradient's magnitude. The impact ratio between two examples is changed only when gradients' magnitude is scaled non-linearly.

2 PRELIMINARIES

We denote a training mini-batch as $\mathbf{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where there are N samples. $(\mathbf{x}_i, y_i)$ represents i-th training sample $\mathbf{x}_i \in \mathbb{R}^D$ and its annotated class label $y_i \in \{1, 2, ..., C\}$. D is the dimensionality of input samples and C is the number of all training classes. Let f_θ be a deep neural network, which transforms $\mathbf{x}_i$ to a representation $\mathbf{f}_i = f_\theta(\mathbf{x}_i) \in \mathbb{R}^E$, E is the dimensionality of target space and θ indicates the parameters to be learned.

To optimise f_θ during training, a linear classifier is generally trained jointly Liu et al. (2016). In general, the linear classifier follows the output embeddings and is composed of one C-neuron fully connected (FC) layer, one softmax normalisation layer and one loss layer. The FC layer can be represented as $\mathbf{z}_i = \mathbf{W}^\top \mathbf{f}_i \in \mathbb{R}^C$, where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_C] \in \mathbb{R}^{E \times C}$ consists of C weight vectors (the bias term is omitted for brevity). $\mathbf{z}_{ij} = \mathbf{w}_j^\top \mathbf{f}_i$ is a logit which indicates the compatibility between sample $\mathbf{x}_i$ and class j. To produce the probabilities of sample $\mathbf{x}_i$ belonging to different classes, we normalise its logit vector $\mathbf{z}_i$ using a softmax function: $p(j|\mathbf{x}_i) = \frac{\exp(\mathbf{z}_{ij})}{\sum_{m=1}^C \exp(\mathbf{z}_{im})}$, where $p(j|\mathbf{x}_i)$ is the probability of sample $\mathbf{x}_i$ being predicted to class j.

Let $q(j|\mathbf{x}_i)$ be the ground-truth probability of $\mathbf{x}_i$ belonging to class j, i.e., $q(j|\mathbf{x}_i) = 1$ if $j = y_i$, $q(j|\mathbf{x}_i) = 0$ otherwise. In the loss layer, if we use CCE, the minimisation objective per iteration is:

$$L_{\mathrm{CCE}}(\mathbf{X}; f_\theta, \mathbf{W}) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C q(j|\mathbf{x}_i) \log p(j|\mathbf{x}_i) = -\frac{1}{N} \sum_{i=1}^N \log p(y_i|\mathbf{x}_i). \tag{1}$$

If MAE is applied, the minimisation objective becomes:

$$L_{\mathrm{MAE}}(\mathbf{X}; f_\theta, \mathbf{W}) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C |p(j|\mathbf{x}_i) - q(j|\mathbf{x}_i)| = \frac{2}{N} \sum_{i=1}^N (1 - p(y_i|\mathbf{x}_i)), \tag{2}$$

where $|\cdot|$ is the absolute function.

In summary, we learn a softmax deep network $g_{\theta,\mathbf{W}}$, which outputs logits: $\mathbf{z}_i = g_{\theta,\mathbf{W}}(\mathbf{x}_i) = \mathbf{W}^\top f_\theta(\mathbf{x}_i) \in \mathbb{R}^C$. In classification tasks, we use $\mathbf{z} = g_{\theta,\mathbf{W}}(\mathbf{x})$ to produce logits for a test image $\mathbf{x}$. While in verification or retrieval tasks Wang et al. (2019a;b;c), we only use $\mathbf{f} = f_\theta(\mathbf{x})$ as an embedding function. The overall pipeline is described in Figure 2. The output of the softmax layer is $\mathbf{p}$.

Definition 1 (Uncertain Examples). *We define uncertain examples to be those data points whose $p(y_i|\mathbf{x}_i)$ are around 0.5. Given an example $\mathbf{x}_i$, if its $p(y_i|\mathbf{x}_i)$ is closer to 0.5, its uncertainty is higher.*

Remark 1. This definition of uncertain examples is intuitive. *If $p(y_i|\mathbf{x}_i)$ is closer to 1, the confidence of $\mathbf{x}_i$ being class y_i is higher. If $p(y_i|\mathbf{x}_i)$ is closer to 0, the confidence of $\mathbf{x}_i$ belonging to one of other classes is higher. However, if $p(y_i|\mathbf{x}_i)$ is around 0.5, we are more uncertain about whether $\mathbf{x}_i$ being class y_i.* Therefore, we can understand uncertainty from the perspective of binary classification (Logistic Regression), i.e., whether $\mathbf{x}_i$ being class y_i or not.

Remark 2. We have *the premise that abnormal (noisy) examples have smaller probabilities in general*. This premise is widely used and demonstrated by our empirical observations. For example, in Figure 4 and Tables 1, 6, the accuracy of noisy subset is less than that of clean subset consistently.

Remark 3. The uncertainty of an example is determined by its probability of being classified to its annotated label. This example can belong to one of the training classes (uncertain in-distribution example), or a class which does not exist in the training set (uncertain out-of-distribution example).

3 GRADIENT MAGNITUDE SERVING AS WEIGHT

As shown in Figure 2, $g_{\theta,\mathbf{W}}$ can be viewed as a black box and the update of θ and $\mathbf{W}$ is based on the back-propagation of logits' gradient. Therefore, an example's contribution can be measured by the magnitude of its partial derivative w.r.t. $\mathbf{z}$. It can be regarded as example weighting that is naturally built-in in loss functions. For brevity and clarity, we summarise the results here and put the detailed derivation in our supplementary material.

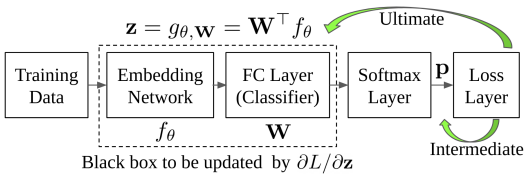


Figure 2: Pipeline of a softmax deep network. There are two reasons for analysing loss functions based on $\frac{\partial L}{\partial \mathbf{z}}$: (1) In gradient back-propagation, the gradients of examples in a mini-batch are fused when computing $\frac{\partial L}{\partial \mathbf{z}}$. (2) Intermediate differences of $\frac{\partial L}{\partial \mathbf{p}}$ lead to ultimate differences of $\frac{\partial L}{\partial \mathbf{z}}$. Therefore, our analysis of $\frac{\partial L}{\partial \mathbf{z}}$ is more direct versus that of $\frac{\partial L}{\partial \mathbf{p}}$ in Zhang & Sabuncu (2018).

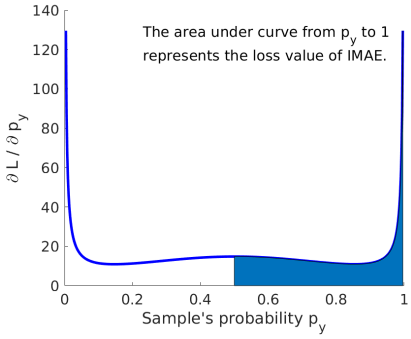


Figure 3: Although the loss expression of IMAE is not an elementary function, we visualise it by integral, i.e., the area under curve from p_y to 1.

3.1 Derivation of Softmax, CCE and MAE Layers

According to $p(j|\mathbf{x}_i)$, Eq. (1), Eq. (2), we have

$$\frac{\partial p(y_i|\mathbf{x}_i)}{\partial \mathbf{z}_{ij}} = \begin{cases} p(y_i|\mathbf{x}_i)(1-p(y_i|\mathbf{x}_i)), & j = y_i \\ -p(y_i|\mathbf{x}_i)p(j|\mathbf{x}_i), & j \neq y_i \end{cases}; \quad \frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)} = \begin{cases} -p(y_i|\mathbf{x}_i)^{-1}, & j = y_i \\ 0, & j \neq y_i \end{cases}; \quad \frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)} = \begin{cases} -2, & j = y_i \\ 0, & j \neq y_i \end{cases}. \tag{3, 4, 5}$$

3.2 Perspective of Derivatives w.r.t. Logits Other Than Probabilities

Prior conclusion according to $\frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)}$, $\frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)}$: Zhang & Sabuncu (2018) concludes that CCE is sensitive to abnormal examples while MAE is robust by treating all data points equally according to Eq. (4) and Eq. (5), respectively.

In this work, we propose to further analyse $\frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{ij}}$, $\frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{ij}}$ as discussed in Figure 2. According to Eq. (3), (4) and (5), we calculate:

$$\frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{ij}} = \begin{cases} p(y_i|\mathbf{x}_i) - 1, & j = y_i \\ p(j|\mathbf{x}_i), & j \neq y_i \end{cases}. \tag{6}$$

$$\frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{ij}} = \begin{cases} 2p(y_i|\mathbf{x}_i)(p(y_i|\mathbf{x}_i) - 1), & j = y_i \\ 2p(y_i|\mathbf{x}_i)p(j|\mathbf{x}_i), & j \neq y_i \end{cases}. \tag{7}$$

Gradient magnitude treated as weight. In CCE and MAE, training samples are weighted because different ones own different gradient magnitude w.r.t. logit vector $\mathbf{z}$. We choose to measure one gradient's magnitude by its L_1 norm because of its simpler statistics than other norms. If one sample's gradient is larger, its impact is larger during gradient back-propagation.

For CCE, based on Eq. (6), the weight of sample $\mathbf{x}_i$ is:

$$w_{\mathrm{CCE}}(\mathbf{x}_i) = ||\frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial \mathbf{z}_i}||_1 = 2(1 - p(y_i|\mathbf{x}_i)), \tag{8}$$

where $||\cdot||_1$ denotes L_1 norm. For MAE, based on Eq. (7), the weight of sample $\mathbf{x}_i$ is:

$$w_{\mathrm{MAE}}(\mathbf{x}_i) = ||\frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_i}||_1 = 4p(y_i|\mathbf{x}_i)(1 - p(y_i|\mathbf{x}_i)). \tag{9}$$

According to Eq. (8) and Eq. (9), in both CCE and MAE, examples' impact is determined by their probabilities being predicted to annotated labels.

4 Improved MAE

IMAE transforms MAE's weighting scheme non-linearly:

$$w_{\mathrm{IMAE}}(\mathbf{x}_i) = \exp(Tp(y_i|\mathbf{x}_i)(1 - p(y_i|\mathbf{x}_i))), \tag{10}$$

where T controls the exponential base. In back-propagation, we simply scale the gradient w.r.t. logits as follows:

$$\frac{\partial L_{\mathrm{IMAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_i} = \frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_i} \frac{w_{\mathrm{IMAE}}(\mathbf{x}_i)}{w_{\mathrm{MAE}}(\mathbf{x}_i)} \Rightarrow ||\frac{\partial L_{\mathrm{IMAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_i}||_1 = w_{\mathrm{IMAE}}(\mathbf{x}_i). \tag{11}$$

IMAE is a family of robust losses when T changes, as summarised in Table 2 and Figure 3.

Table 2: Summary of CCE, MAE and IMAE. $(\mathbf{x}, y)$ is a training example. For simplicity, $p_y = p(y|\mathbf{x})$, and $p_j = p(j|\mathbf{x}), j \neq y$, $\sum_{j \neq y} p_j + p_y = 1$. Prior analysis on loss functions is based on the loss expression or $\frac{\partial L}{\partial \mathbf{p}}$. Instead, we are the first to study the differences of loss functions according to $||\frac{\partial L}{\partial \mathbf{z}}||_1$. Our empirical evidences justifies its rationality. Note that we have $L(p_y) = \int \frac{\partial L}{\partial p_y} dp_y, L(1) = 0$, therefore $L(p_y) = \int_{p_y}^{1} -\frac{\partial L}{\partial p_y} dp_y$. We remark ***IMAE is neither symmetric nor bounded***, which challenges the robustness theories studied in Ghosh et al. (2017); Zhang & Sabuncu (2018); Wang et al. (2019d).

	Loss Expression $L = L(p_y) = \int_{p_y}^{1} -\frac{\partial L}{\partial p_y} dp_y$	$\frac{\partial L}{\partial \mathbf{p}}$		$\frac{\partial p_y}{\partial \mathbf{z}}$		$\frac{\partial L}{\partial \mathbf{z}} = \sum_{j=1}^{C} \frac{\partial L}{\partial p_j} \times \frac{\partial p_j}{\partial \mathbf{z}}$		$\|\|\frac{\partial L}{\partial \mathbf{z}}\|\|_1$
		$\frac{\partial L}{\partial p_y}$	$\frac{\partial L}{\partial p_j}, j \neq y$	$\frac{\partial p_y}{\partial \mathbf{z}_y}$	$\frac{\partial p_y}{\partial \mathbf{z}_j}, j \neq y$	$\frac{\partial L}{\partial \mathbf{z}_y}$	$\frac{\partial L}{\partial \mathbf{z}_j}, j \neq y$	
CCE	$-\log p_y$	$-\frac{1}{p_y}$	0	$p_y(1-p_y)$	$-p_y p_j$	$p_y - 1$	p_j	$2(1-p_y)$
MAE	$2(1-p_y)$	-2	0	$p_y(1-p_y)$	$-p_y p_j$	$2p_y(p_y-1)$	$2p_y p_j$	$4p_y(1-p_y)$
IMAE	$\int_{p_y}^{1} \frac{\exp(Tp_y(1-p_y))}{2p_y(1-p_y)} dp_y$	$\frac{\exp(Tp_y(1-p_y))}{2p_y(p_y-1)}$	0	$p_y(1-p_y)$	$-p_y p_j$	$\frac{\exp(Tp_y(1-p_y))}{-2}$	$\frac{\exp(Tp_y(1-p_y))p_j}{2(1-p_y)}$	$\exp(Tp_y(1-p_y))$

4.1 Design Motivation: To Adjust Gradient Magnitude's Variance and Impact Ratio

Linear scaling also changes magnitude variance. However, it cannot adjust impact ratio, i.e., the ratio between two gradients' magnitude. That is why we have tried linear scaling and find it does not work.

Instead, the exponential function is non-linear so that the impact ratio of one sample versus another is re-adjusted compared with original MAE. The hyper-parameter T controls how significant gradient magnitude's variance and impact ratio are changed.

Furthermore, assuming that samples' probabilities are uniformly distributed, we compute the gradients' variance of MAE and IMAE over training data points:

$$\sigma_{\text{MAE}} = \int_0^1 w_{\text{MAE}}^2(p)\,\mathrm{d}p - (\int_0^1 w_{\text{MAE}}(p)\,\mathrm{d}p)^2 \tag{12}$$

$$\sigma_{\text{IMAE}} = \int_0^1 w_{\text{IMAE}}^2(p)\,\mathrm{d}p - (\int_0^1 w_{\text{IMAE}}(p)\,\mathrm{d}p)^2. \tag{13}$$

We have $\sigma_{\text{MAE}} = 0.09$. When $T = 8$, $\sigma_{\text{IMAE}} = 4.55$.

4.2 Discussion of MAE and CCE

The weighting curves of CCE, MAE and IMAE are compared in Figure 1. Our key findings are summarised in the end of introduction. We further discuss them as follows:

- MAE's weighting scheme is appealing and practical in that samples with medium probabilities are emphasized. Generally, high-probability samples are clean and already trained well. While low-probability ones are highly likely to be noisy as a model improves during training. Although all samples are not trained well and probabilities are not meaningful at the beginning, it also does not hurt to focus on medium-probability ones.
- MAE's gradient magnitude's variance over data points is only 0.09. As a consequence, the impact ratio of one example versus another is too small. Therefore, the majority contribute almost equally. Therefore, MAE generally underfits to training data.
- Does high loss value usually back-propagate high gradients to update parameters? The answer is NO. Therefore, those theorems based on loss values, e.g., symmetric or bounded conditions are insufficient for analysing robustness of DNNs Ghosh et al. (2017). Actually, ***IMAE is neither symmetric nor bounded***. However, it is proved to be noise-robust empirically.

These analytical discussions are demonstrated in our empirical studies in Table 6 and Figures 4, 10.

5 EXPERIMENTS

We demonstrate the effectiveness of IMAE as follows:

Outperforming the state-of-the-art. IMAE is compared with recent baselines in Sections 5.1 and 5.2 in different scenarios: (1) Clean labels; (2) Synthetic symmetric and asymmetric noisy labels; (3) Realistic agnostic noise.

Analysis of the training dynamics of IMAE against CCE and MAE. We thoroughly visualise and compare the training dynamics of IMAE, CCE and MAE in Section 5.3 for empirical justification.

Supplementary studies. In our supplementary material, we further prove IMAE's effectiveness by: (1) The results on a *video retrieval task* (video person re-identification); (2) The results of *different stochastic optimisers*; (3) *The ablation study of* T.

5.1 IMAGE CLASSIFICATION ON CIFAR-100 WITH SYNTHETIC NOISE

Dataset. CIFAR-100 Krizhevsky (2009) contains 100 classes, 500 images per class for training and 100 images per class for testing. The image size is 32×32.

Synthetic label noise generation. (1) Class-independent (uniform or symmetric) noise: With a probability of r, the label of each image is replaced by one of the other class labels uniformly. (2) Class-dependent (non-uniform or asymmetric) noise: The 100 classes of CIFAR-100 are grouped into 20 coarse ones. Every coarse one has 5 fine classes. Following Wang et al. (2019d), we first randomly select 2 out of 5 classes, and then their labels are flipped to each other with a probability of r. r denotes the noise rate. All instances generated from the same original image by data augmentation share the same label. All test labels are kept intact.

Implementation details. We follow the settings of recent SL Wang et al. (2019d) and train ResNet44 He et al. (2016) for a fair comparison with their reported results. We also use the same data augmentation techniques: random horizontal flips and crops of 32×32 on the images after being padded with 4 pixels on each side. All networks are trained using SGD with a momentum of 0.9, a weight decay of 0.0005 and an initial learning rate of 0.1.

Baselines. IMAE is compared against standard CCE, MAE, and six recent robust training baselines: 1) Forward (or Backward) applies a noise-transition matrix to multiply the network's predictions (or losses) for label correction purpose (Patrini et al., 2017); 2) Bootstrapping learns on new labels generated by a convex combination (soft or hard combinations) of the original ones and their predictions (Reed et al., 2015). 3) D2L achieves noise-robustness by restricting the dimensionality expansion of learned subspaces during training (Ma et al., 2018); 4) SL boosts CCE with a noise-robust counterpart, i.e., reverse cross entropy (Wang et al., 2019d); 5) GCE aims to achieve a balance between MAE and CCE Zhang & Sabuncu (2018); 6) Label Smoothing (LS) trains DNNs on softly smoothed labels instead of one-hot ones; We remark that Lee et al. (2019) is not benchmarked for two reasons: (1) The used network is not ResNet-44 by checking with the authors; (2) The proposed algorithm is orthogonal to ours because it targets at the inference stage and is a generative classifier on top of pre-trained deep representations. Our IMAE focuses on the training stage and is a softmax-based neural classifier.

Results. We display the results in Tables 3 and 4. We observe that IMAE is superior to the state-of-the-art. We fix the random seed as 123 and do not use any random computational accelerator for the purpose of exact reproducibility.

5.2 IMAGE CLASSIFICATION ON CLOTHING1M WITH REALISTIC UNKNOWN NOISE

Dataset. Clothing1M Xiao et al. (2015) contains one million clothing images of fourteen classes from online shopping websites. *Its noise type is agnostic.* The noise rate is around 38.46%. Additionally, it includes 50k, 14k, and 10k images with clean labels for training, validation, and testing, respectively. To compare fairly with existing algorithms without exploiting auxiliary information from trusted clean data, we also train only on the noisy training data.

Implementation details. We follow Patrini et al. (2017); Tanaka et al. (2018); Wang et al. (2019d) and train ResNet50 initialised by pretrained ImageNet model Russakovsky et al. (2015). We apply an

Table 3: The results on CIFAR-100 using ResNet44. Results from SL and D2L are different due to different optimisation details. In our experiments, we fix the random seed as 123 and do not use any random computational accelerator for the purpose of exact reproducibility. The best results on each block and our IMAE are bolded.

	Method	Clean Labels	Symmetric Noisy Labels		
			r=0.2	r=0.4	r=0.6
Results From SL	CCE	64.3	59.3	50.8	25.4
	LS	63.7	58.8	50.1	24.7
	Boot-hard	63.3	57.9	48.2	12.3
	Forward	64.0	59.8	53.1	24.7
	D2L	64.6	59.2	52.0	35.3
	GCE	64.4	59.1	53.3	36.2
	SL	**66.8**	**60.0**	**53.7**	**41.5**
Results From D2L	CCE	68.2	52.9	42.9	30.1
	Boot-hard	68.3	58.5	44.4	36.7
	Boot-soft	67.9	57.3	41.9	32.3
	Forward	68.5	60.3	51.3	41.2
	Backward	68.5	58.7	45.4	34.5
	D2L	**68.6**	**62.2**	**52.0**	**42.3**
Our Trained Results	CCE	**70.0**	60.4	53.2	42.1
	MAE	8.2	6.4	7.3	5.2
	IMAE	**69.2**	**63.4**	**54.7**	**43.9**

Table 4: The results on CIFAR-100 using ResNet44. The best results on each block are bolded.

	Method	Asymmetric Noisy Labels		
		r=0.2	r=0.3	r=0.4
Results From SL Wang et al. (2019d)	CCE	63.0	63.1	61.9
	LS	63.0	62.3	61.6
	Bootstrap	63.4	63.2	62.1
	Forward	64.1	64.0	60.9
	D2L	62.4	63.2	61.4
	GCE	63.0	63.2	61.7
	SL	**65.6**	**65.1**	**63.1**
Our trained Results	CCE	66.4	64.7	60.3
	MAE	7.3	6.3	7.3
	IMAE	**67.5**	**65.8**	**63.3**

Table 5: Classification accuracy (%) on Clothing1M with ResNet50 He et al. (2016). The leftmost block's results are from SL Wang et al. (2019d) while the middle block's are from Masking Han et al. (2018).

CCE	Boot-hard	Forward	D2L	GCE	SL	S-adaptation	Masking	Joint Optim.	Our trained results		
									CCE	MAE	IMAE
68.8	68.9	69.8	69.5	69.8	71.0	70.3	71.1	72.2	71.7	39.7	**73.2**

SGD optimiser with a momentum of 0.9 and a weight decay of 0.00002. We set the initial learning rate to 0.01 and divide it by 10 after 10k and 15k iterations. We stop training at 30k iterations. Regarding data augmentation, a raw input image is warped to 256×256, followed by a random crop of 227×227 and a random horizontal mirroring. The batch size is 84. Every program is run on a single Tesla V100 GPU with 32 GB RAM.

Competitors. Some recent baselines are compared: 1) S-adaptation explicitly estimates latent true labels by an additional softmax layer Goldberger & Ben-Reuven (2017); 2) Masking speculates the structure of a noise-transition matrix with human cognition Han et al. (2018); 3) Joint Optim. iteratively optimises model's parameters and latent true labels Tanaka et al. (2018). Others are

introduced in Section 5.1. Note that Han et al. (2019) corrects labels gradually and Li et al. (2019) exploits meta-learning. They are not technically related and not benchmarked consequently.

Results. We display the results in Table 5. IMAE outperforms the state-of-the-art, which proves IMAE's effectiveness under real-world scenarios with agnostic noise. Beyond, we remark that IMAE is much simpler than those competitors except CCE, MAE.

5.3 Empirical Analysis of IMAE Against Basic Baselines CCE and MAE on CIFAR-10

Dataset. CIFAR-10 Krizhevsky (2009) contains 10 classes, 5k images per class for training and 1k images per class for testing. The image size is 32×32.

Implementation details [3]**.** We follow the study on CIFAR-10 in He et al. (2016), which means we use exactly the same architectures (ResNet20, ResNet56) and training settings: a weight decay of 0.0001, a momentum of 0.9, a batch size of 128. The learning rate starts at 0.1, then is divided by 10 at 32k and 48k iterations. Training stops at 100k iterations. Data augmentation is the same as CIFAR-100. For IMAE, *without tuning T case by case*, we fix $T = 0.5$ when training data is clean and $T = 8$ when noise exists although noisy rate is different.[4]

A well-accepted way to improve data fitting ability is increasing a model's capacity. Therefore, we train a shallower net ResNet20 and a deeper net ResNet56 for better analysis.

5.3.1 CIFAR-10 with intact labels

In Table 6, we first compare IMAE with CCE and MAE on clean CIFAR-10 using different nets (ResNet20, ResNet56). We observe that IMAE is competitive with CCE and outperforms MAE significantly.

5.3.2 CIFAR-10 with corrupted labels

Following Zhang et al. (2017); Arpit et al. (2017), we test the robustness of deep models against corrupted labels. We evaluate on uniform noise because it is more challenging than asymmetric noise which is verified in Vahdat (2017).

Majority voting assumption. When generating uniform noise on CIFAR-10, even up to 80% noise rate, clean examples are still the majority because 80% labels are corrupted to other 9 classes evenly. *We remark that the majority voting is our reasonable assumption.* We believe that if the noise becomes the majority, it is hard to discover meaningful patterns. Being natural and intuitive, the majority define the meaningful data patterns to learn.

Results. The results are summarised in Table 6. For more comprehensive and clear comparison, we display the training dynamics in Figures 4 (40% noise) and 10 (80% noise) of the supplementary material. *Note that general learning objectives are high final testing accuracy, low accuracy on the noisy training subset, and high accuracy on the clean training subset.* Therefore, we report the hybrid accuracy on the combination of testing set and clean training set. We have the following observations:

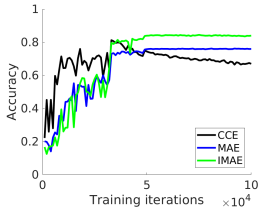
- Regarding CCE's test accuracies, the best is always much higher than the final. In Figures 4 and 10, *as training goes, CCE always tries to fit the noisy training subset better.* Therefore, CCE learns a lot of error information when severe noise exists. When it comes to MAE and IMAE, the gap between the best and final accuracies is significantly smaller than that of CCE regardless of net's capacity.
- The training accuracies on both noisy and clean subsets are compared. Whatever the noise rate and net's capacity are, CCE fits the noisy subset much more. Although MAE fits the noisy subset much less, it fits the clean subset worst. Instead, our IMAE fits the noisy subset little and the clean subset competitively with CCE.
- IMAE obtains the best hybrid accuracy consistently.

[3]Our purpose is to study the behaviours of CCE, MAE and IMAE on CIFAR-10 instead of pushing its state-of-the-art results.

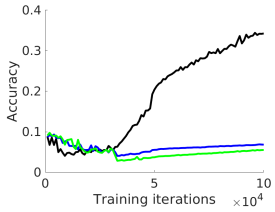
[4]More discussion about the hyper-parameter T is given in our supplementary material.

Table 6: Results (%) of CCE, MAE and IMAE on CIFAR-10 with different noise rates. For classification accuracy on the testing set, we show the best result achieved during training and the final result when training stops, which are indicated by 'Best' and 'Final', respectively. For training accuracy, the results on noisy and clean subsets are displayed. The hybrid accuracy represents the result on the combination of testing set and clean training set. We report training and hybrid accuracies of the final model when training terminates. *The ultimate objective is to achieve high hybrid accuracy, since both training and testing data points may occur in a deployed system.* The best result on each column block is bolded. '–' indicates there is no noisy subset.

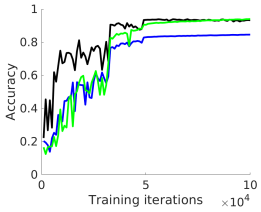
Backbone	Noise rate	Loss	Testing accuracy		Training accuracy: Naive fitting		Hybrid accuracy: Meaningful patterns
			Best	Final	Noisy subset	Clean subset	
ResNet20	0%	CCE	91.5	91.3	–	**100**	**98.5**
		MAE	89.3	89.2	–	95.8	94.7
		IMAE	**91.7**	**91.4**	–	99.8	98.4
	40%	CCE	81.2	67.0	34.3	93.3	72.6
		MAE	76.2	75.9	6.8	84.6	79.7
		IMAE	**84.3**	**84.0**	**5.5**	**94.0**	**88.2**
	80%	CCE	43.0	20.3	38.3	57.0	22.0
		MAE	27.7	27.5	**9.7**	29.4	27.8
		IMAE	**52.0**	**41.0**	16.8	**64.8**	**41.5**
ResNet56	0%	CCE	**92.4**	**92.2**	–	**100**	**98.7**
		MAE	89.0	89.0	–	96.1	94.9
		IMAE	92.2	**92.2**	–	99.8	98.5
	40%	CCE	81.6	63.3	75.0	**96.2**	63.6
		MAE	67.0	66.9	8.1	74.3	70.2
		IMAE	**82.2**	**81.5**	**6.5**	93.1	**86.5**
	80%	CCE	**38.2**	16.4	52.5	**62.3**	18.7
		MAE	15.2	15.1	**9.6**	15.6	15.1
		IMAE	37.1	**34.0**	13.0	44.7	**34.8**



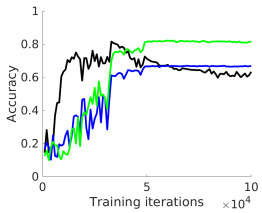
(a) ResNet20: Testing set (higher is better).



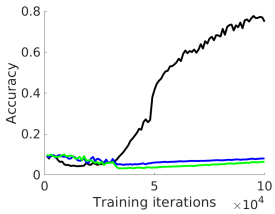
(b) ResNet20: Noisy subset (lower is better).



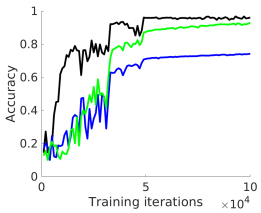
(c) ResNet20: Clean subset (higher is better).



(d) ResNet56: Testing set (higher is better).



(e) ResNet56: Noisy subset (lower is better).



(f) ResNet56: Clean subset (higher is better).

Figure 4: CIFAR-10 with noise rate $r = 40\%$. The accuracies on testing set, noisy subset and clean subset of training set along with training iterations. The legend on the top left is shared by all subfigures. *Better viewed in colour.*

6 Related Work

IMAE is a family of robust loss functions, inspired by the intrinsic example weighting scheme of MAE. Therefore, our work is related to some prior work about example weighting and robust loss functions.

6.1 EXAMPLE WEIGHTING

In Ren et al. (2018), a meta-learning algorithm weights data points according to their gradient directions. The meta-learning algorithm is optimised on a clean validation set. In contrast, our IMAE assigns weights to samples based on their gradient magnitude and does not require extra clean set. MentorNet Jiang et al. (2018) learns data-driven weighting scheme, which guides StudentNet to focus on samples whose labels are more trustful. In Active Bias Chang et al. (2017) and Focal Loss Lin et al. (2017), uncertain and hard examples are emphasised, respectively. Other related work on weighting samples includes curriculum learning Bengio et al. (2009), self-paced learning Kumar et al. (2010), and hard examples mining Shrivastava et al. (2016); Wang et al. (2019b). *In summary, what makes ours special is that the weighting scheme inherits from MAE, which is naturally built-in in the loss function without intuitive designing.*

6.2 NOISE-ROBUST THEOREMS ON LOSS FUNCTIONS

Noise-robust theorems on loss functions from the angle of symmetric and bounded conditions on loss values have been studied recently Ghosh et al. (2017); Zhang & Sabuncu (2018); Wang et al. (2019d). Does a robust loss function have to be symmetric or bounded? The answer is NO according to this work. Although IMAE is neither symmetric nor bounded, we have extensive empirical studies to support its effectiveness.

7 CONCLUSION

We firstly present a thorough study of CCE and MAE technically and empirically. Compared with previous work, we introduce our observations and new conclusions: 1) MAE underfits to meaningful patterns; 2) MAE is noise-tolerant because of emphasising on medium-probability (uncertain) examples instead of treating all samples equally. Secondly, we claim gradient magnitude's variance matters. As a consequence, we propose an effective and simple solution for addressing MAE's underfitting issue while preserving its noise-robustness. IMAE is a family of robust loss functions whose gradient magnitude's variance is adjustable.

We remark that our empirically demonstrated claim–"Gradient Magnitude's Variance Matters"–can be applied for other algorithms as well, for example, CCE. However, it is beyond the scope of this work since we focus on analysing MAE and how to improve MAE here. We will investigate this claim in other loss functions in our future work.

Furthermore, we have a research plan to study the effectiveness of IMAE's variants for the robustness against adversarial perturbations Kurakin et al. (2017); Moosavi-Dezfooli et al. (2017), e.g., incorporating IMAE's variants with iterative trimmed loss minimisation Shen & Sanghavi (2019).

REFERENCES

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, 2017.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Smooth loss functions for deep top-k classification. *ICLR*, 2018.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*, 2017.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, 2018.

Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

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 *ICML*, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NeurIPS*, 2010.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017.

Marc T Law, Raquel Urtasun, and Richard S Zemel. Deep spectral clustering learning. In *ICML*, 2017.

Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *ICML*, 2019.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, 2019.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

Yu Liu, Junjie Yan, and Wanli Ouyang. Quality aware network for set to set recognition. In *CVPR*, 2017.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017.

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 *CVPR*, 2017.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR Workshop*, 2015.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pp. 211–252, 2015.

Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *ICML*, 2019.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.

Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *NeurIPS*, 2017.

Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. *CVPR*, 2019a.

Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, and Neil M. Robertson. Deep metric learning by online soft mining and class-aware attention. In *AAAI*, 2019b.

Xinshao Wang, Elyor Kodirov, Yang Hua, and Neil M Robertson. Id-aware quality for set-based person re-identification. *arXiv preprint arXiv:1911.09143*, 2019c.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, 2019d.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.

Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *ECCV*, 2016.

Supplementary Material for IMAE

A THE IMPACT OF T ON GRADIENT MAGNITUDE'S VARIANCE

Assuming samples' probabilities are uniformly distributed, we calculate the variances of IMAE's weighting curves with different T. As illustrated in Sec. 4 of the main paper, we rewrite the Eq. (10) (We use e to replace $\exp$ for brevity):

$$w_{\mathrm{IMAE}}(p) = e^{T \cdot p(1-p)}, \tag{14}$$

where p is the probability of one randomly sampled example being predicted to its annotated label. According to Eq. (13) in the main paper, we have,

$$\begin{aligned} \sigma_{\mathrm{IMAE}} &= \int_0^1 w_{\mathrm{IMAE}}^2(p)\,\mathrm{d}p - (\int_0^1 w_{\mathrm{IMAE}}(p)\,\mathrm{d}p)^2 \\ &= \int_0^1 \mathrm{e}^{2Tp(1-p)}\,\mathrm{d}p - (\int_0^1 \mathrm{e}^{Tp(1-p)}\,\mathrm{d}p)^2 \\ &= \frac{\sqrt{\pi}\,\mathrm{erf}\left(\frac{\sqrt{2T}}{2}\right) e^{\frac{T}{2}}}{\sqrt{2T}} - \frac{\pi\,\mathrm{erf}^2\left(\frac{\sqrt{T}}{2}\right) e^{\frac{T}{2}}}{T}. \end{aligned} \tag{15}$$

erf is the error function. Therefore we obtain the weighting variances σ_{IMAE} of IMAE with different T, as displayed in Table 7.

Table 7: The weight variance (gradient magnitude's variance) of IMAE when T changes.

T	16	8	4	2	1	0.5	0
σ_{IMAE}	354.113	4.546	0.299	0.040	0.007	0.002	0

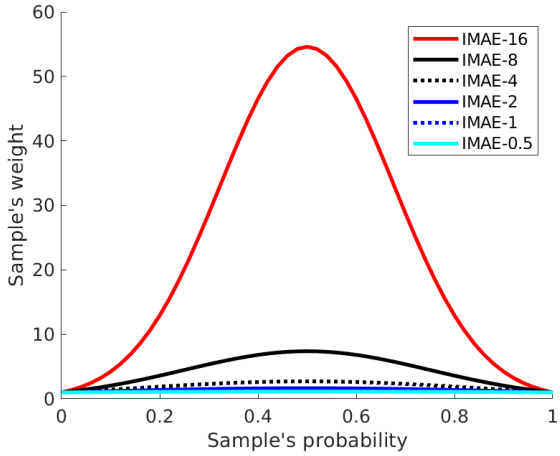
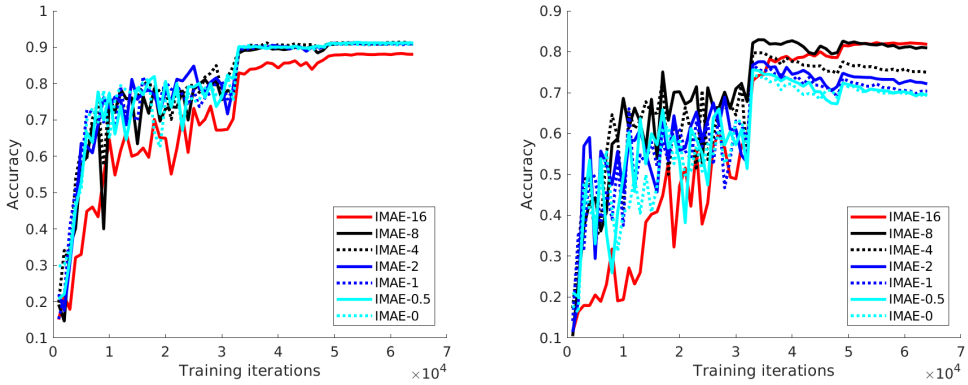


Figure 5: Sample's weight along with probability in IMAE with different T (IMAE-T). The hyper-parameter T controls gradient magnitude's variance, and impact ratio between examples consequently. *Better viewed in colour.*

B THE IMPACT OF T ON VALIDATION ACCURACY

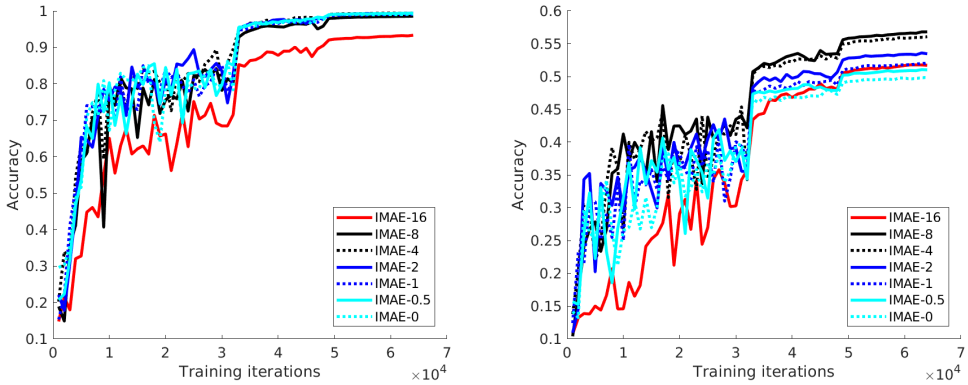
We visualise and compare the effect of T on CIFAR-10 test performance. These experiments follow exactly the same settings of the main paper.



(a) Test accuracies of IMAE-T trained on intact training labels.

(b) Test accuracies of IMAE-T trained on corrupted training labels.

Figure 6: The accuracy on CIFAR-10 *test set* along with training iterations. We display the results when training on intact training set and corrupted training set. *Better viewed in colour.*



(a) The training accuracies of IMAE-T on intact training set.

(b) The training accuracies of IMAE-T on corrupted training set.

Figure 7: The accuracy on CIFAR-10 *training sets* along with training iterations. We show the results when training on intact training set and corrupted training set. *Better viewed in colour.*

We try two cases: (1). Training labels are intact ($r = 0$); (2). Training labels are corrupted randomly with a probability of 0.4 ($r = 40\%$). In both cases, the test set is kept intact for evaluation. The backbone network is ResNet20.

B.1 CIFAR-10 WITH INTACT TRAINING LABELS

The test results are shown and compared in Figure 6a.

When training labels are clean, it is unhelpful to differentiate training samples in a high degree, e.g., the performance is even lower when $T = 16$. The final test accuracies are similar when T ranges from 0 to 8.

B.2 CIFAR-10 WITH CORRUPTED TRAINING LABELS

The results are presented and compared in Figure 6b. Because there exists 40% label noise, as training goes, the test accuracy drops, which means the model overfits noisy data gradually.

However, **we observe that higher differentiation degree (larger T) works better and is much less susceptible to overfitting to noisy data.** In Figure 6b, the final test accuracies of IMAE-16 and IMAE-8 are much higher than those of other models.

C THE IMPACT OF T ON TRAINING ACCURACY

Following the practice in the main paper, we also visualise and compare the accuracies on the training sets, which indicate how different models fit to training data as training goes, thus leading to different generalisation performance in the test phase. We present how each model fits its corresponding training set in Figure 7.

C.1 FITTING OF INTACT TRAINING SET

As compared in Figure 7a, **all models fit training data similarly when T ranges from 0 to 8**. However, when $T = 16$, the differentiation degree becomes too large as shown in Table 7. When differentiation degree is too large, only a quite small proportion of training data can contribute. Consequently, **IMAE-16 underfits training data compared with other models.** That is why IMAE-16 has the worst test performance as shown in Figure 6a.

C.2 FITTING OF CORRUPTED TRAINING SET

The training accuracies of corrupted training set are displayed in Figure 7b. We have two observations:

- **In cases where noise rate is high, as T increases, the fitting of training data first becomes better, and then becomes worse.** Specifically, when T increases from 0 to 8, the training accuracy grows gradually, which means the fitting of training data becomes better. However, when $T = 16$, the weighting variance becomes very large (Table 7). As a result, IMAE-16's fitting of training data becomes much worse than IMAE-8's.
- **Fitting corrupted training data better does not mean better generalisation performance.** On the one hand, although IMAE-16 fits the training data much worse than IMAE-8 (Figure 7b), IMAE-16's test accuracy is slightly better than IMAE-8's (Figure 6b). On the other hand, similar to IMAE-8, IMAE-4 fits its training data well (Figure 7b), but IMAE-4's test performance is much worse than IMAE-8's (Figure 6b).

D CHOOSING T IN PRACTICE

In summary, the training accuracy (fitting of training data) is uninformative for estimating a model's generalisation performance according to our findings in Section C.2. Therefore, it is better to optimise T on a validation set in practice.

For empirical demonstration, since the overlap rate between corrupted and intact training sets is only $(1 - r) = 60\%$, *we treat the original intact training set as a validation set*. The validation performance of IMAE-16, IMAE-8 and IMAE-4 is compared in Figure 8. We observe that IMAE-16 and IMAE-8 own similar validation performance, while IMAE-4's validation accuracy is lower. Furthermore, their validation performance is consistent with their test performance (Figure 6b). Therefore, we conclude that it is a good practice to optimise T on a validation set in different cases.

E VIDEO PERSON RE-IDENTIFICATION

Dataset and evaluation settings. MARS contains 20,715 videos of 1,261 persons Zheng et al. (2016). There are 1,067,516 frames in total. Because person videos are collected by tracking and detection algorithms, abnormal examples exist as shown in Figure 9. The exact noise rate is unknown. Following standard settings, we use 8,298 videos of 625 persons for training and 12,180 videos of other 636 persons for testing. We report the cumulated matching characteristics (CMC) and mean average precision (mAP) results.

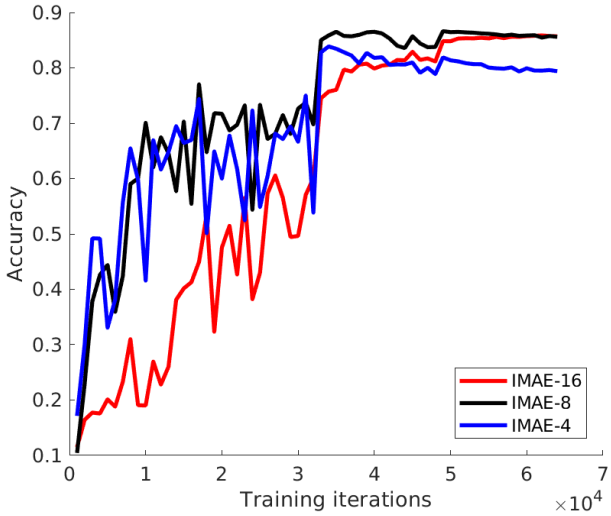


Figure 8: IMAE-16's, IMAE-8's and IMAE-4's accuracies on the clean training set when they are trained on the corrupted training set. The overlap rate between corrupted and intact training sets is only $(1 - r) = 60\%$. Therefore, *we can use the original training set as a validation set. Better viewed in colour.*

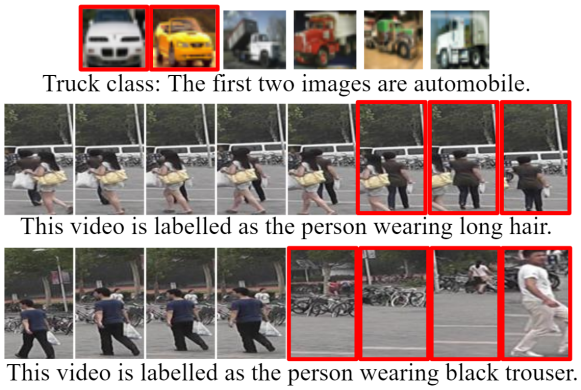


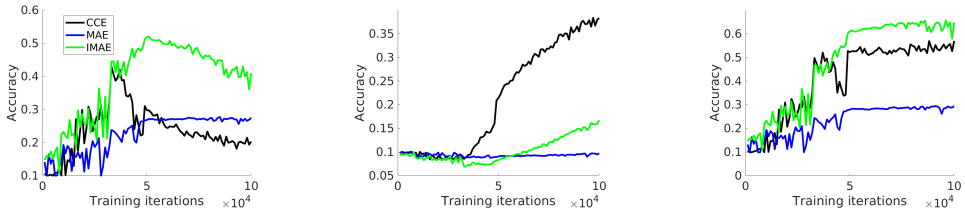
Figure 9: Display of abnormal training examples highlighted by red boxes. The 1st row shows synthetic abnormal examples from corrupted CIFAR-10 Krizhevsky (2009). The 2nd and 3rd rows present realistic abnormal examples from video person re-identification benchmark MARS Zheng et al. (2016). We remark three representatives: 1) The abnormal images with no person in 3rd row contain no semantic information at all. 2) The last abnormal image in 2nd or 3rd row may contain a person that does not belong to any person in the training set. 3) We cannot decide the object of interest without any prior when an image contains more than one object, e.g., the 2nd and 3rd last images in 2nd row contain two persons. *Better viewed in colour.*

Implementation details.[5] Following Liu et al. (2017); Wang et al. (2019b), we train GoogleNet V2. We also treat a video as an image set, which means we use only appearance information without exploiting latent temporal information. A video's representation is simply the average fusion of its frames' representations. *We apply the same training settings for each loss*. The learning rate starts from 0.01 and is divided by 2 every 10k iterations. We stop training at 50k iterations. We choose SGD optimiser with a weight decay of 0.0005 and momentum of 0.9. The batch size is set to 180. We use standard data augmentation: a 227×227 crop is randomly sampled and flipped after resizing an original image to 256×256. At testing, following Wang et al. (2019b); Movshovitz-

[5]We explore the performance of different losses in real-world applications instead of pushing the state-of-the-art results.

Table 8: The results of algorithms using different stochastic optimisers on CIFAR-10 with 40% class-independent (symmetric) label noise. The trained network is ResNet56 He et al. (2016). The key hyper-parameters of all optimisers are shown. Other settings are fixed to be the same as presented in the implementation details of Section 5.3, e.g., weight decay = 0.0001. Since Adam is an adaptive gradient method, we show several variants of it.

	SGD (lr: 0.01)	SGD + Momentum (lr: 0.01)	Nesterov (lr: 0.01)	Adam (lr: 0.01, delta: 0.1)	Adam (lr: 0.005, delta: 0.1)	Adam (lr: 0.005, delta: 1)
CCE	64.3	60.6	56.4	42.5	44.5	50.3
MAE	39.3	64.7	64.1	68.2	59.9	41.4
GCE	68.8	80.5	79.7	73.2	70.6	69.3
IMAE	**82.0**	**83.5**	**83.7**	**75.5**	**76.3**	**78.6**



(a) ResNet20: Testing set (higher is better). (b) ResNet20: Noisy subset (lower is better). (c) ResNet20: Clean subset (higher is better).

(d) ResNet56: Testing set (higher is better). (e) ResNet56: Noisy subset (lower is better). (f) ResNet56: Clean subset (higher is better).

Figure 10: CIFAR-10 with noise rate $r = 80\%$. The accuracies on testing set, noisy subset and clean subset of training set along with training iterations. The legend on the top left is shared by all subfigures. *Better viewed in colour.*

Attias et al. (2017); Law et al. (2017), we first L_2 normalise videos' features and then calculate the cosine similarity between every two features.

Results. We compare our method with CCE, MAE and GCE. We implement GCE with its best settings. The results are shown in Table 9. IMAE outperforms other related methods by a significant margin.

Table 9: The retrieval results of CCE, MAE, GCE and IMAE on MARS with GoogLeNet V2 Ioffe & Szegedy (2015).

Metric	CCE	MAE	GCE	IMAE
mAP (%)	58.1	12.0	31.6	**70.9**
CMC-1 (%)	73.8	26.0	51.5	**83.5**

F THE RESULTS OF IMAE USING DIFFERENT STOCHASTIC OPTIMISERS

In this section, we study the performance of IMAE when different stochastic optimisers are used. The results are presented in Table 8. We observe that IMAE's results are the best consistently.

G DERIVATION OF SOFTMAX, CCE AND MAE LAYERS

G.1 DERIVATION OF SOFTMAX LAYER

As the softmax layer is shared by CCE and MAE, we present the derivation of softmax layer first. First, we have

$$p(y_i|\mathbf{x}_i)^{-1} = 1 + \sum_{j \neq y_i} \exp(\mathbf{z}_{ij} - \mathbf{z}_{iy_i}). \tag{16}$$

If $j = y_i$, for left and right sides of Eq. (16), we calculate their derivatives w.r.t. $\mathbf{z}_{iy_i}$ simultaneously:

$$\begin{aligned} \frac{-1}{p(y_i|\mathbf{x}_i)^2} \frac{\partial p(y_i|\mathbf{x}_i)}{\mathbf{z}_{iy_i}} &= -\sum_{j \neq y_i} \exp(\mathbf{z}_{ij} - \mathbf{z}_{iy_i}) \\ => \frac{\partial p(y_i|\mathbf{x}_i)}{\mathbf{z}_{iy_i}} &= p(y_i|\mathbf{x}_i)(1 - p(y_i|\mathbf{x}_i)). \end{aligned} \tag{17}$$

If $j \neq y_i$, analogously we have:

$$\begin{aligned} \frac{-1}{p(y_i|\mathbf{x}_i)^2} \frac{\partial p(y_i|\mathbf{x}_i)}{\mathbf{z}_{ij}} &= \exp(\mathbf{z}_{ij} - \mathbf{z}_{iy_i}) \\ => \frac{\partial p(y_i|\mathbf{x}_i)}{\mathbf{z}_{ij}} &= -p(y_i|\mathbf{x}_i)p(j|\mathbf{x}_i). \end{aligned} \tag{18}$$

In summary, the derivation of softmax layer is:

$$\frac{\partial p(y_i|\mathbf{x}_i)}{\partial \mathbf{z}_{ij}} = \begin{cases} p(y_i|\mathbf{x}_i)(1 - p(y_i|\mathbf{x}_i)), & j = y_i \\ -p(y_i|\mathbf{x}_i)p(j|\mathbf{x}_i), & j \neq y_i \end{cases} \tag{19}$$

G.2 DERIVATION OF LOSS LAYER: CCE

According to Eq. (1), we have

$$L_{\mathrm{CCE}}(\mathbf{x}_i; f_\theta, \mathbf{W}) = -\log p(y_i|\mathbf{x}_i). \tag{20}$$

Therefore, we obtain (the parameters are omitted for brevity),

$$\frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)} = \begin{cases} -p(y_i|\mathbf{x}_i)^{-1}, & j = y_i \\ 0, & j \neq y_i \end{cases}. \tag{21}$$

G.3 DERIVATION OF LOSS LAYER: MAE

According to Eq. (2), we have

$$L_{\mathrm{MAE}}(\mathbf{x}_i; f_\theta, \mathbf{W}) = 2(1 - (p(y_i|\mathbf{x}_i)). \tag{22}$$

Therefore, we obtain

$$\frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)} = \begin{cases} -2, & j = y_i \\ 0, & j \neq y_i \end{cases}. \tag{23}$$

G.4 DERIVATIVES W.R.T. $\mathbf{z}_i$

$\partial L_{\mathrm{CCE}}(\mathbf{x}_i)/\partial \mathbf{z}_i$: The calculation is based on Eq. (21) and Eq. (19).

If $j = y_i$, we have:

$$\begin{aligned} \frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{iy_i}} &= \sum_{j=1}^{C} \frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)} \frac{\partial p(y_i|\mathbf{x}_i)}{\mathbf{z}_{iy_i}} \\ &= p(y_i|\mathbf{x}_i) - 1. \end{aligned} \tag{24}$$

If $j \neq y_i$, it becomes:

$$\begin{aligned}\frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{ij}} &= \sum_{j=1}^{C} \frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)} \frac{\partial p(y_i|\mathbf{x}_i)}{\mathbf{z}_{ij}} \\ &= p(j|\mathbf{x}_i).\end{aligned} \tag{25}$$

In summary, $\partial L_{\mathrm{CCE}}(\mathbf{x}_i)/\partial \mathbf{z}_i$ can be represented as:

$$\frac{\partial L_{\mathrm{CCE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{ij}} = \begin{cases} p(y_i|\mathbf{x}_i) - 1, & j = y_i \\ p(j|\mathbf{x}_i), & j \neq y_i \end{cases}. \tag{26}$$

$\partial L_{\mathrm{MAE}}(\mathbf{x}_i)/\partial \mathbf{z}_i$: The calculation is analogous with that of $\partial L_{\mathrm{CCE}}(\mathbf{x}_i)/\partial \mathbf{z}_i$. According to Eq. (23) and Eq. (19), if $j = y_i$:

$$\begin{aligned}\frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{iy_i}} &= \sum_{j=1}^{C} \frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)} \frac{\partial p(y_i|\mathbf{x}_i)}{\mathbf{z}_{iy_i}} \\ &= -2p(y_i|\mathbf{x}_i)(1 - p(y_i|\mathbf{x}_i)).\end{aligned} \tag{27}$$

otherwise ($j \neq y_i$):

$$\begin{aligned}\frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{ij}} &= \sum_{j=1}^{C} \frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial p(j|\mathbf{x}_i)} \frac{\partial p(y_i|\mathbf{x}_i)}{\mathbf{z}_{ij}} \\ &= 2p(y_i|\mathbf{x}_i)p(j|\mathbf{x}_i).\end{aligned} \tag{28}$$

In summary, $\partial L_{\mathrm{MAE}}(\mathbf{x}_i)/\partial \mathbf{z}_i$ is:

$$\frac{\partial L_{\mathrm{MAE}}(\mathbf{x}_i)}{\partial \mathbf{z}_{ij}} = \begin{cases} 2p(y_i|\mathbf{x}_i)(p(y_i|\mathbf{x}_i) - 1), & j = y_i \\ 2p(y_i|\mathbf{x}_i)p(j|\mathbf{x}_i), & j \neq y_i \end{cases}. \tag{29}$$