A Data-Driven Measure of Relative Uncertainty for Misclassification Detection

Eduardo Dadalto[*]
Laboratoire des signaux et systèmes (L2S)
Université Paris-Saclay CNRS CentraleSupélec
91190 Gif-sur-Yvette, France
eduardo.dadalto@centralesupelec.fr

Marco Romanelli[*]
New York University
New York, NY, USA
mr6852@nyu.edu

Georg Pichler[*]
Institute of Telecommunications
TU Wien
1040 Vienna, Austria
georg.pichler@ieee.org

Pablo Piantanida
International Laboratory on Learning Systems (ILLS)
Quebec AI Institute (MILA)
CNRS, CentraleSupélec - Université Paris-Saclay
pablo.piantanida@cnrs.fr

Abstract

Misclassification detection is an important problem in machine learning, as it allows for the identification of instances where the model's predictions are unreliable. However, conventional uncertainty measures such as Shannon entropy do not provide an effective way to infer the real uncertainty associated with the model's predictions. In this paper, we introduce a novel data-driven measure of uncertainty relative to an observer for misclassification detection. Interestingly, according to the proposed measure, soft-predictions that correspond to misclassified instances can carry a large amount of uncertainty, even though they may have low Shannon entropy. We demonstrate improvements over multiple image classification tasks, outperforming state-of-the-art misclassification detection methods.

1 Introduction

Critical applications, such as autonomous driving and automatic tumor segmentation, have benefited greatly from machine learning algorithms. This motivates the importance of understanding their limitations and urges the need for methods that can detect patterns on which the model uncertainty may lead to dangerous consequences [1]. A recent thread of research addresses misclassifications by augmenting the training data for better representation [23, 22, 15]. However, in order to build the detectors, these approaches rely on some statistics of the posterior distribution output by the model, e.g., the entropy, interpreting it as an expression of the model's confidence. Regrettably, these measures suffer from two major inconveniences: they are invariant to relabeling of the underlying label space, and, more importantly, they lead to very low uncertainty values for overconfident predictions, even if they are wrong, making them unfit for the purpose of detection of misclassification instances.

In this work, we propose a data-driven measure of relative uncertainty inspired by [16]. By learning to minimize the uncertainty on positive instances and to maximize it on negative instances, our metric can effectively capture meaningful information to differentiate between the underlying structure of distributions corresponding to two categories of data. **Our measure is "relative", as it is not characterized axiomatically, but only serves the purpose of measuring uncertainty of positive instances relative to negative ones from the point of view of a subjective observer d. Our contributions** are three-fold:

[*]Equal contribution.

Mathematics of Modern Machine Learning Workshop at NeurIPS 2023.

1. We leverage a novel statistical framework for categorical distributions to devise a learnable measure of relative uncertainty (REL-U) for a model's predictions, which induces large uncertainty for negative instances, even if they may lead to low Shannon entropy;
2. We propose a closed-form solution for training REL-U in the presence of positive and negative instances;
3. We report significantly favorable and consistent results over different models and datasets, considering both natural misclassifications within the same statistical population, and in case of distribution shift, or *mismatch*, between training and testing distributions.

2 From Uncertainty to Misclassification Detection

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a (possibly continuous) feature space and let $\mathcal{Y} = \{1, \dots, C\}$ denote the label space related to some task of interest. Moreover, we denote by p_{XY} the underlying probability density function (pdf) on $\mathcal{X} \times \mathcal{Y}$. We assume that a machine learning model is trained on some training data, which ultimately yields a model that, given features $\mathbf{x} \in \mathcal{X}$, outputs a probability mass function (pmf) on $\mathcal{Y}$, which we denote as a vector $\hat{\mathbf{p}}(\mathbf{x})$. This may result from a soft-max output layer, for example. A predictor $f\colon \mathcal{X} \to \mathcal{Y}$ is then constructed, which yields $f(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \hat{\mathbf{p}}(\mathbf{x})_y$. We note that we may also interpret $\hat{\mathbf{p}}(\mathbf{x}) \in [0,1]^{|\mathcal{Y}|}$ as the probability distribution of $\hat{Y}$, which, given $\mathbf{X} = \mathbf{x}$, is distributed according to $p_{\hat{Y}|\mathbf{X}}(y|\mathbf{x}) \triangleq \hat{\mathbf{p}}(\mathbf{x})_y$.

We define the indicator of the misclassification event as $E(\mathbf{X}) \triangleq \mathbb{1}[f(\mathbf{X}) \neq Y]$. The occurrence of the "misclassification" event is then characterized by $E = 1$. Misclassification detection is a standard binary classification problem, where E needs to be estimated from $\mathbf{X}$. We will denote the misclassification detector as $g\colon \mathcal{X} \to \{0,1\}$. The underlying pdf p_X can be expressed as a mixture of two random variables: $\mathbf{X}_+ \sim p_{X|E}(\mathbf{x}|0)$ (positive instances) and $\mathbf{X}_- \sim p_{X|E}(\mathbf{x}|1)$ (negative instances), where $p_{X|E}(\mathbf{x}|1)$ and $p_{X|E}(\mathbf{x}|0)$ represent the pdfs conditioned on the error event and the event of correct classification, respectively.

3 A Data-Driven Measure of Uncertainty

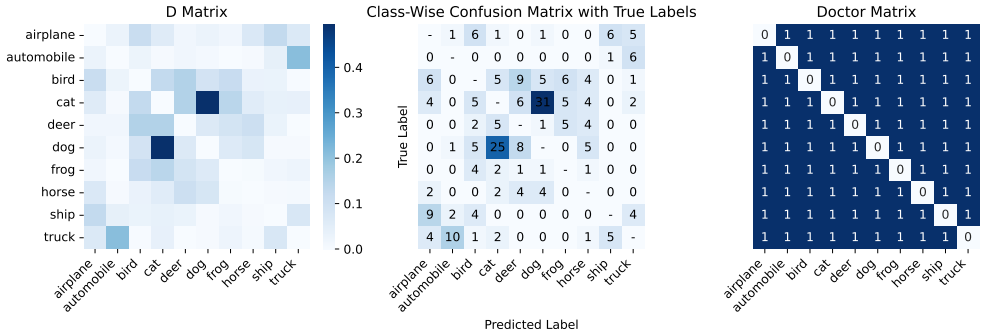


Figure 1: Intuitive example illustrating the advantage of REL-U compared to entropy-based methods: REL-U (left-end side heatmap) captures the real uncertainty (central heatmap) much better than Doctor [6]; a detailed analysis is provided in Section 4.

In stark contrast with measures of information uncertainty such as Shannon entropy [19, Sec. 6], Rényi entropy [18], q-entropy [20], as well as several divergence measures, capturing a notion of distance between probability distributions, such as Kullback-Leibler divergence [12], f-divergence [3], and Rényi divergence [18], we propose a notion of "relative" uncertainty that is not invariant w.r.t. relabeling of the underlying label space, thus preserving the semantic meaning of the labels.

We propose to construct a class of uncertainty measures which is inspired by the measure of diversity investigated in [16]. Recall that the quantity $\hat{\mathbf{p}}(\mathbf{x})$ is the posterior distribution output by the model given the input $\mathbf{x}$. Let $s\colon \mathcal{X} \to \mathbb{R}$ be the uncertainty measure in (1) that assigns a score $s(\mathbf{x})$ to every

feature $\mathbf{x}$ in the input space $\mathcal{X}$ defined as

$$s_d(\mathbf{x}) \triangleq \mathbb{E}[d(\widehat{Y}, \widehat{Y}')|\mathbf{X} = \mathbf{x}] = \sum_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} d(y, y') \hat{\mathbf{p}}(\mathbf{x})_y \hat{\mathbf{p}}(\mathbf{x})_{y'}, \tag{1}$$

where $d \in \mathcal{D}$ is in a class of distance measures and, given $\mathbf{X} = \mathbf{x}$, the random variables $\widehat{Y}, \widehat{Y}' \sim \hat{\mathbf{p}}(\mathbf{x})$ are independently and identically distributed according to $\hat{\mathbf{p}}(\mathbf{x})$. We can derive a misclassification detector g by fixing a threshold $\gamma \in \mathbb{R}$, $g(\mathbf{x}; s, \gamma) = \mathbb{1}[s(\mathbf{x}) \leq \gamma]$, where $g(\mathbf{x}) = 1$ when $E = 1$.

The statistical framework we are introducing here offers great flexibility by allowing for an arbitrary function d that can be learned from data, as opposed to fixing a predetermined distance as in [16]. **In essence, we regard the uncertainty in equation 1 as relative to a given observer d, which appears as a parameter in the definition.** To the best of our knowledge, this is a fundamentally novel concept of uncertainty.

We first rewrite $s_d(\mathbf{x})$ (1) in order to make it amenable to learning the metric d. By defining the $C \times C$ matrix $D \triangleq (d_{ij})$ using $d_{ij} = d(i, j)$, we have $s_d(\mathbf{x}) = \hat{\mathbf{p}}(\mathbf{x})\, D\, \hat{\mathbf{p}}(\mathbf{x})^\top$. For $s_d(\mathbf{x})$ to yield a good detector g, we design a contrastive objective, where we would like $\mathbb{E}[s_d(\mathbf{X}_+)]$, which is the expectation over the positive samples, to be small compared to the expectation over negative samples, i.e., $\mathbb{E}[s_d(\mathbf{X}_-)]$. This naturally yields to the following objective function, where we assume the usual properties of a distance function $d(y, y) = 0$ and $d(y', y) = d(y, y') \geq 0$ for all $y, y' \in \mathcal{Y}$.

Definition 1. *Let us introduce our objective function with hyperparameter $\lambda \in [0, 1]$,*

$$\mathcal{L}(D) \triangleq (1 - \lambda) \cdot \mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_+)\, D\, \hat{\mathbf{p}}(\mathbf{X}_+)^\top\right] - \lambda \cdot \mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_-)\, D\, \hat{\mathbf{p}}(\mathbf{X}_-)^\top\right] \tag{2}$$

and for a fixed $K \in \mathbb{R}^+$, define our optimization problem as follows:

$$\begin{cases} \text{minimize}_{D \in \mathbb{R}^{C \times C}}\ \mathcal{L}(D) & \\ \textit{subject to} & d_{ii} = 0, \qquad \forall i \in \mathcal{Y} \\ & d_{ij} \geq 0, \qquad \forall i, j \in \mathcal{Y} \\ & d_{ij} = d_{ji}, \qquad \forall i, j \in \mathcal{Y} \\ & \text{Tr}(DD^\top) \leq K \end{cases} \tag{3}$$

The first constraint in equation 3 states that the elements along the diagonal are zeros, which ensures that the uncertainty measure is zero when the distribution is concentrated at a single point. The second constraint ensures that all elements are non-negative, which is a natural condition so the measure of uncertainty is non-negative. The natural symmetry between two elements stems from the third constraint, while the last constraint imposes a constant upper-bound on the Frobenius norm of the matrix D, guaranteeing that a solution for the underlying learning problem exists.

Proposition 1 (Closed form solution)**.** *The constrained optimization problem defined in* (3) *admits a closed form solution $D^* = \frac{1}{Z}(d^*_{ij})$, where*

$$d^*_{ij} = \begin{cases} \text{ReLU}\left(\lambda \cdot \mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_-)_i^\top \hat{\mathbf{p}}(\mathbf{X}_-)_j\right] - (1 - \lambda) \cdot \mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_+)_i^\top \hat{\mathbf{p}}(\mathbf{X}_+)_j\right]\right) & i \neq j \\ 0 & i = j \end{cases}. \tag{4}$$

The multiplicative constant Z is chosen such that D^* satisfies the condition $\text{Tr}(D^*(D^*)^\top) = K$.

The proof is based on a Lagrangian approach and relegated to Appendix A.1. Finally, we define the Relative Uncertainty (REL-U) score for a given feature $\mathbf{x}$ as

$$s_{\text{REL-U}}(\mathbf{x}) \triangleq \hat{\mathbf{p}}(\mathbf{x})\, D^*\, \hat{\mathbf{p}}(\mathbf{x})^\top. \tag{5}$$

Remark. *Note that the Gini coefficint $H_2(\widehat{Y}|\mathbf{x}) = -\log \sum_{y \in \mathcal{Y}} \left(\hat{\mathbf{p}}(\mathbf{x})_y\right)^2$ proposed in [6] is a special case of* (5) *when $d_{ij} = 1$ if $i \neq j$ and $d_{ii} = 0$. Thus, $s_{1-d}(\mathbf{x}) = s_{\text{gini}}(\mathbf{x})$ when choosing d to be the Hamming distance, which was also pointed out in [16, Note 1].*

4 Experiments and Discussion

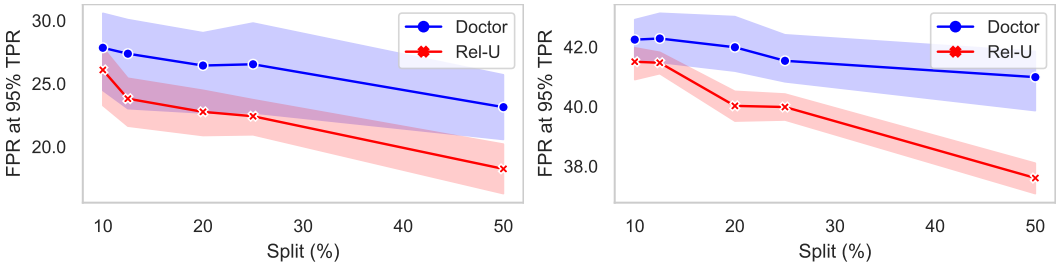
In this section, we present the experiments conducted to validate our measure of uncertainty in the context of misclassification considering both the case when the training and test distributions *match*,

and the case in which the two distributions *mismatch*. Although our method requires additional positive and negative instances, we show that lower amounts are needed (hundreds or few thousands) compared to methods that involve re-training or fine-tuning (hundreds of thousands).

For a given model architecture and dataset, we split the test set into two: one portion for tuning the our method and beselines and the other for evaluating it. Consequently, we can compute all *hyperparameters* in an unbiased way and cross-validate performance over many splits generated from ten random seeds. For details on temperature and input pre-processing, see Appendix A.4. As of *evaluation metric*, we consider the false positive rate (fraction of misclassifications detected as being correct classifications) when 95% of data is true positive (fraction of correctly classified samples detected as being correct classifications), denoted as FPR at 95% TPR (lower is better). This metric is commonly used in the literature of misclassification and out-of-distribution detection [9]. Our main results are reported in Table 1 , Table 2 in Appendix A.4, and Figure 2. We observed gains in FPR and the AUROC results are similar among methods (see Figure 3 in the appendix). In Appendix A.4, we ablate on the impact of each hyperparameter, studies the impact of calibration, and detection performance on inputs with corrupted covariates or belonging to novel classes.

Table 1: Misclassification detection performance in terms of average FPR at 95% TPR (lower is better) in percentage with one standard deviation over ten different seeds in parenthesis.

Model	Training	Accuracy	MSP	ODIN	Doctor	REL-U
ResNet-34 (CIFAR-10)	CrossEntropy	95.4	25.8 (4.8)	19.4 (1.0)	14.3 (0.2)	**14.1** (0.1)
	LogitNorm	94.3	30.5 (1.6)	**26.0** (0.6)	31.5 (0.5)	31.3 (0.6)
	Mixup	96.1	60.1 (10.7)	38.2 (2.0)	26.8 (0.6)	**19.0** (0.3)
	OpenMix	94.0	40.4 (0.0)	39.5 (1.3)	**28.3** (0.7)	28.5 (0.2)
	RegMixUp	97.1	34.0 (5.2)	26.7 (0.1)	21.8 (0.2)	**18.2** (0.2)
ResNet-34 (CIFAR-100)	CrossEntropy	79.0	42.9 (2.5)	38.3 (0.2)	34.9 (0.5)	**32.7** (0.3)
	LogitNorm	76.7	58.3 (1.0)	55.7 (0.1)	**65.5** (0.2)	**65.4** (0.2)
	Mixup	78.1	53.5 (6.3)	43.5 (1.6)	**37.5** (0.4)	**37.5** (0.3)
	OpenMix	77.2	46.0 (0.0)	43.0 (0.9)	41.6 (0.3)	**39.0** (0.2)
	RegMixUp	80.8	50.5 (2.8)	45.6 (0.9)	40.9 (0.8)	**37.7** (0.4)



(a) CIFAR-10 (b) CIFAR-100

Figure 2: Impact of the tuning split size on the misclassification performance on a ResNet-34 model trained with supervised CE loss for our method and the Doctor. Hyperparameters are set to default values ($T = 1.0$, $\epsilon = 0.0$, and $\lambda = 0.5$), so that only the impact of the validation split size is observed.

Empirical Interpretation of the Relative Uncertainty Matrix. Figure 1 exemplifies the advantage of our method over the entropy-based methods. In particular, the left-end side heatmap represents the D matrix learned by optimizing (2) on CIFAR-10. Clearly, by only using the information required in (2) (no class labels or predictions required, only the probability vectors), our method is able to describe the uncertainty over different, and differently hard to predict, classes: darker shades of blue indicate higher uncertainty, while lighter shades of blue indicate lower uncertainty. The central heatmap is the predictor's class-wise true confusion matrix. The vertical axis represents the true class, while the horizontal axis represents the predicted class. For each combination of two classes ij, the corresponding cell reports the count of samples of class j that were predicted as class i. The correct matches along the diagonal are dashed for better visualization of the mistakes. The confusion matrix is computed on the same validation set used to compute the D matrix. Crucially, our uncertainty

matrix can express different degrees of uncertainty depending on the specific combination of classes at hand. Let us focus for instance on the fact that most of the incorrectly classified dogs are predicted as cats, and vice-versa. Our matrix D fully captures this by assigning high uncertainty to the cells at the intersection between these two classes. Conversely, entropy-based methods assign the same uncertainty to all the cells, regardless of the specific combination of classes at hand.

5 Summary and Concluding Remarks

In this paper, we propose a **method for uncertainty assessment that departs from the conventional practice of directly measuring uncertainty through the entropy of the output distribution**. REL-U uses a metric that leverages higher uncertainty score for negative data w.r.t. positive data, e.g., incorrectly and correctly classified samples in the context of misclassification detection, and attains favorable results on matched and mismatched data. In addition, our method stands out for its *flexibility and simplicity*, as it relies on a closed form solution to an optimization problem.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] Imre Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizität von markoffschen ketten. *Magyer Tud. Akad. Mat. Kutato Int. Koezl.*, 8:85–108, 1964.

[4] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 7068–7081, 2021.

[5] Corrado Gini. Variabilità e mutabilità; contributo allo studio delle distribuzioni e delle relazioni statistiche. In *[Fasc. I.]. Tipogr. Di P. Cuppini 1912.*, 1912.

[6] Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. DOCTOR: A simple method for detecting misclassification errors. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 5669–5681, 2021.

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[9] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[10] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017.

[11] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[12] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.

[13] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[14] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Ann Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[15] Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip Torr, and Puneet K. Dokania. Regmixup: Mixup as a regularizer can surprisingly improve accuracy and out distribution robustness. In *Advances in Neural Information Processing Systems*, 2022.

[16] C Radhakrishna Rao. Diversity and dissimilarity coefficients: a unified approach. *Theoretical population biology*, 21(1):24–43, 1982.

[17] Jie Ren, Stanislav Fort, Jeremiah Z. Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *CoRR*, abs/2106.09022, 2021.

[18] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*. Berkeley, California, USA, 1961.

[19] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

[20] Constantino Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of statistical physics*, 52:479–487, 1988.

[21] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, 2022.

[22] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017.

[23] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Openmix: Exploring outlier samples for misclassification detection. *ArXiv*, abs/2303.17093, 2023.

[24] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Rethinking confidence calibration for failure prediction. *ArXiv*, abs/2303.02970, 2023.

A Appendix

A.1 Proof of Proposition 1

We have the optimization problem

$$\begin{cases} \text{minimize}_{D\in\mathbb{R}^{C\times C}}\ \mathcal{L}(D) & \\ \text{subject to} & d_{ii}=0,\ \forall i\in\{1,\dots,C\}; \\ & d_{ij}-d_{ji}=0,\ \forall i,j\in\{1,\dots,C\} \\ & \operatorname{Tr}(DD^\top)-K\leq 0 \\ & -d_{ij}\leq 0,\ \forall i,j\in\{1,\dots,C\} \end{cases} \tag{6}$$

in standard form [2, eq. (4.1)] and can thus apply the KKT conditions [2, eq. (5.49)]. We find

$$\nabla\mathcal{L}(D^*)-\sum_{i,j}\xi^*_{ij}\nabla d^*_{ij}+\sum_i\mu^*_i\nabla d^*_{ii}+\sum_{ij}\nu^*_{ij}\nabla(d^*_{ij}-d^*_{ji})+\kappa^*\nabla(\operatorname{Tr}(D^*(D^*)^\top)-K)=0 \tag{7}$$

as well as the constraints

$$d^*_{ii}=0 \qquad d^*_{ij}-d^*_{ji}=0 \tag{8}$$

$$-d^*_{ij}\leq 0 \qquad \xi^*_{ij}\geq 0 \tag{9}$$

$$\xi^*_{ij}d_{ij}=0 \qquad \kappa^*\geq 0 \tag{10}$$

$$\kappa^*(\operatorname{Tr}(D^*(D^*)^\top)-K)=0 \tag{11}$$

We have

$$\nabla\mathcal{L}(D^*)=(1-\lambda)\cdot\mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_+)^\top\hat{\mathbf{p}}(\mathbf{X}_+)\right]-\lambda\cdot\mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_-)^\top\hat{\mathbf{p}}(\mathbf{X}_-)\right] \tag{12}$$

$$\nabla(\operatorname{Tr}(D^*(D^*)^\top)-K)=2D^* \tag{13}$$

and thus[2]

$$\begin{aligned} 0=&(1-\lambda)\cdot\mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_+)^\top\hat{\mathbf{p}}(\mathbf{X}_+)\right]-\lambda\cdot\mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_-)^\top\hat{\mathbf{p}}(\mathbf{X}_-)\right]-\boldsymbol{\xi}^*+\operatorname{diag}(\boldsymbol{\mu}^*) \\ &+\boldsymbol{\nu}^*-(\boldsymbol{\nu}^*)^\top+\kappa^*2D^* \end{aligned} \tag{14}$$

$$\begin{aligned} D^*=\frac{1}{2\kappa^*}\Big(&-(1-\lambda)\cdot\mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_+)^\top\hat{\mathbf{p}}(\mathbf{X}_+)\right]+\lambda\cdot\mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_-)^\top\hat{\mathbf{p}}(\mathbf{X}_-)\right]+\boldsymbol{\xi}^*-\operatorname{diag}(\boldsymbol{\mu}^*) \\ &-\boldsymbol{\nu}^*+(\boldsymbol{\nu}^*)^\top\Big) \end{aligned} \tag{15}$$

As $\nabla\mathcal{L}(D^*)$ in (12) is already symmetric, we can choose $\boldsymbol{\nu}^*=\mathbf{0}$. We choose[3] $\boldsymbol{\mu}^*=\operatorname{diag}(\nabla\mathcal{L}(D^*))$ to ensures $d^*_{ii}=0$. The non-negativity constraint can be satisfied by appropriately choosing $\mathbf{0}\leq\boldsymbol{\xi}^*=\operatorname{ReLU}(-\nabla\mathcal{L}(D^*))$. Finally, κ^* is chosen such that the constraint $\operatorname{Tr}(D^*(D^*)^\top)=K$ is satisfied. In total, this yields $D^*=\frac{1}{Z}\operatorname{ReLU}(d^*_{ij})$, where

$$d^*_{ij}=\begin{cases}-(1-\lambda)\cdot\mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_+)_i^\top\hat{\mathbf{p}}(\mathbf{X}_+)_j\right]+\lambda\cdot\mathbb{E}\left[\hat{\mathbf{p}}(\mathbf{X}_-)_i^\top\hat{\mathbf{p}}(\mathbf{X}_-)_j\right] & i\neq j\\ 0 & i=j\end{cases}. \tag{16}$$

The multiplicative constant $Z=2\kappa^*>0$ is chosen such that D^* satisfies the condition $\operatorname{Tr}(D^*(D^*)^\top)=K$.

Remark. *A technical problem may occur when d^*_{ij} as defined in* (16) *is equal to zero for all $i,j\in\{1,2,\dots,C\}$. In this case, D^* cannot be normalized to satisfy $\operatorname{Tr}(D^*(D^*)^\top)=K$ and the solution to the optimization problem in* (6) *is the all-zero matrix $D^*=\mathbf{0}$. I.e., no learning is performed in this case. We deal with this problem by falling back to the Gini coefficient* (17)*, where similarly no learning is required, where the Gini coefficient [5] is defined as:*

$$s_{\text{gini}}(\mathbf{x})\triangleq 1-\sum_{y\in\mathcal{Y}}\left(\hat{\mathbf{p}}(\mathbf{x})_y\right)^2 \tag{17}$$

[2]We use $\mathbf{X}=\operatorname{diag}(\mathbf{x})$ for a vector $\mathbf{x}$ to obtain a matrix $\mathbf{X}$ with $\mathbf{x}$ on the diagonal and zero otherwise.

[3]Slightly abusing notation, we also write $\mathbf{x}=\operatorname{diag}(\mathbf{X})$ to obtain the diagonal of the matrix $\mathbf{X}$ as a vector $\mathbf{x}$.

Equivalently, one may also add a small numerical correction ε to the definition of the ReLU *function, i.e.,* $\overline{\mathrm{ReLU}}(x) = \max(x, \varepsilon)$. *Using this slightly adapted definition when defining* $D^* = \frac{1}{Z}\overline{\mathrm{ReLU}}(d^*_{ij})$ *naturally yields the Gini coefficient in this case.*

A.2 Limitations.

We presented machine learning researchers with a fresh methodological outlook and provided machine learning practitioners with a user-friendly tool that promotes safety in real-world scenarios. Some considerations should be put forward, such as the importance of cross-validating the hyperparameters of the detection methods to ensure their robustness on the targeted data and model. As a data-driven measure of uncertainty, to achieve the best performance, it is important to have enough samples at the disposal to learn the metric from. As every detection method, our method may be vulnerable to targeted attacks from malicious users.

A.3 Temperature Scaling an Input Pre-Processing

Temperature scaling involves the use of a scalar coefficient $1/T \in \mathbb{R}^+$ that is multiplied by the logits of the network before computing the softmax. This has an effect on the network confidence and the posterior output probability distribution. The final temperature-scaled-softmax function is given by:

$$\sigma(z) = \frac{\exp\left(z/T\right)}{\sum_j \exp\left(z_j/T\right)}.$$

Moreover, the perturbation is applied to the input image in order to increase the network "sensitivity" to the input. In particular, the perturbation is given by:

$$x' = x - \varepsilon \times \operatorname{sign}\left[-\nabla_x \log\left(s_{\text{REL-U}}(\mathbf{x})\right]\right,$$

for $\varepsilon > 0$.

A.4 Extra Results on Misclassification Detection

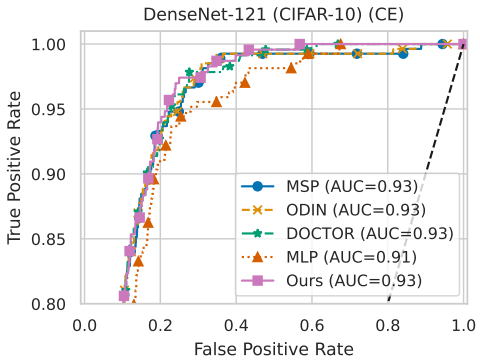
Table 1 showcases the misclassification detection performance in terms of FPR at 95% TPR of our method and the strongest baselines (MSP [9], ODIN [13], Doctor [6]) on different neural network architectures (DenseNet-121 [10], ResNet-34 [8]) trained on different datasets (CIFAR-10, CIFAR-100 [11]) with different learning objectives (Cross Entropy loss, LogitNorm [21], MixUp [22], RegMixUp [15], OpenMix [23]). We observe that, on average, our method performs best 11/20 experiments and is equal to the second best in 4/9 out of the remaining experiments. It works consistently better on all the models trained with cross-entropy loss and the models trained with RegMixUp objective, which achieved the best accuracy among them. We observed some negative results when training with logit normalization, but also, the accuracy of the base model decreases. Results on Bayesian methods and an MLP directly trained on the tuning data are reported to Table 3 together with additional results in the Appendix A.4.

The performance of REL-U is comparable to other methods in terms of AUROC while outperforming them in high-TPR regions and reducing the risk of classification errors when abstention is desired (coverage) as observed in Figure 3.

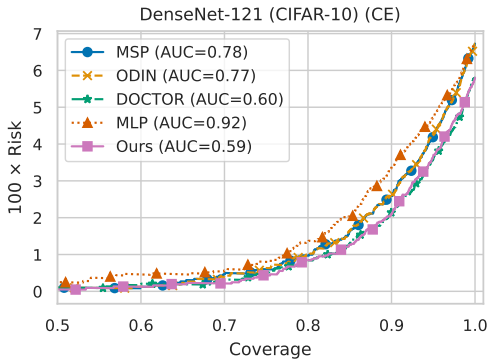
We also compared our method to Bayesian baselines and an MLP network trained on the same data we used to tune our method. Their results are reported in Table 3.

Ablation study. Figure 2 displays how the amount of data reserved for the tuning split impacts the performance of the best two detection methods. We demonstrate how our data-driven uncertainty estimation metric generally improves with the amount of data fed to it in the tuning phase, especially on a more challenging setup such as on CIFAR-100 model. Figure 4 illustrates three ablation studies conducted to analyze and comprehend the effects of different factors on the experimental results. A separate subplot represents each hyperparameter ablation study, showcasing the outcomes obtained under specific conditions. **We observe that $\lambda \geq 0.5$, low temperatures, and low noise magnitude achieve better performance.** Overall, the method is shown to be robust to the choices of hyperparameters under reasonable ranges.

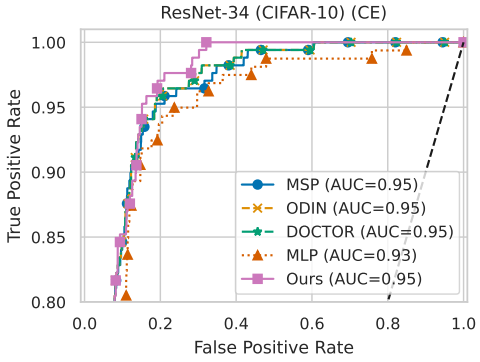
Does calibration improves detection? There has been growing interest in developing machine learning algorithms that are not only accurate but also well-calibrated, especially in applications



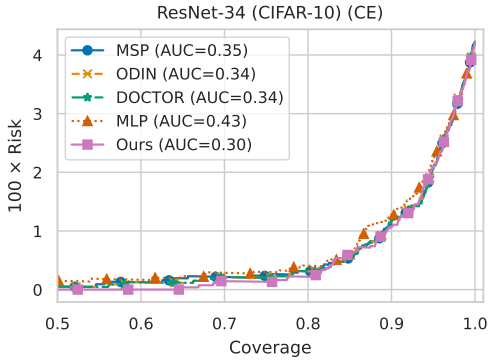
(a) DenseNet-121 ROC curve.



(b) DenseNet-121 RC curve.



(c) ResNet-34 ROC curve.



(d) ResNet-34 RC curve.

Figure 3: Equivalent performance of the detectors in terms of ROC demonstrating lower FPR for our method for high TPR regime. The risk and coverage (RC) curves also looks similar between methods, with a small advantage to our method in terms of AURC.

Table 2: Misclassification detection results across two different architectures trained on CIFAR-10 and CIFAR-100 with five different training losses. We report the average accuracy of these models and the detection performance in terms of average FPR at 95% TPR (lower is better) in percentage with one standard deviation over ten different seeds in parenthesis.

Model	Training	Accuracy	MSP	ODIN	Doctor	REL-U
DenseNet-121 (CIFAR-10)	CrossEntropy	94.0	32.7 (4.7)	24.5 (0.7)	21.5 (0.2)	**18.3** (0.2)
	LogitNorm	92.4	39.6 (1.2)	**32.7** (1.0)	37.4 (0.5)	37.0 (0.4)
	Mixup	95.1	54.1 (13.4)	38.8 (1.2)	**24.5** (1.9)	37.6 (0.9)
	OpenMix	94.5	57.5 (0.0)	53.7 (0.2)	33.6 (0.1)	**31.6** (0.4)
	RegMixUp	95.9	41.3 (8.0)	30.4 (0.4)	23.3 (0.4)	**22.0** (0.2)
DenseNet-121 (CIFAR-100)	CrossEntropy	73.8	45.1 (2.0)	41.7 (0.4)	**41.5** (0.2)	**41.5** (0.2)
	LogitNorm	73.7	66.4 (2.4)	**60.8** (0.2)	68.2 (0.4)	68.0 (0.4)
	Mixup	77.5	48.7 (2.3)	41.4 (1.4)	**37.7** (0.6)	**37.7** (0.6)
	OpenMix	72.5	52.7 (0.0)	51.9 (1.3)	48.1 (0.3)	**45.0** (0.2)
	RegMixUp	78.4	49.7 (2.0)	45.5 (1.1)	43.3 (0.4)	**40.0** (0.2)

where reliable probability estimates are desirable. In this section, we investigate whether models with calibrated probability predictions help improve the detection capabilities of our method or not. Previous work [24] has shown that calibration does not particularly help or impact misclassification detection on models with similar accuracies, however, they focused only on calibration methods and overlooked detection methods.

Table 3: Misclassification detection results across two different architectures trained on CIFAR-10 and CIFAR-100 with CrossEntropy loss. We report the detection performance in terms of average FPR at 95% TPR (lower is better) in percentage with one standard deviation over ten different seeds in parenthesis.

Model	Dataset	MC-Dropout	Ensemble	MLP	REL-U
DenseNet-121	CIFAR-10	30.3 (3.8)	25.5 (0.8)	37.3 (5.8)	**18.3** (0.2)
DenseNet-121	CIFAR-100	47.6 (1.2)	45.9 (0.7)	78.4 (1.4)	**41.5** (0.2)
ResNet-34	CIFAR-10	25.8 (4.9)	14.8 (1.4)	33.6 (2.7)	**14.1** (0.1)
ResNet-34	CIFAR-100	42.3 (1.0)	37.4 (1.9)	63.3 (1.0)	**32.7** (0.3)

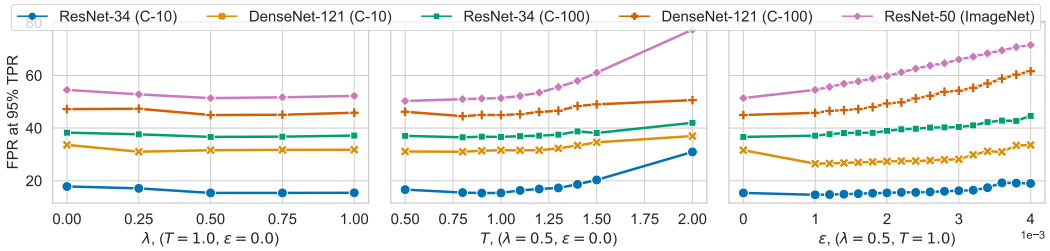


Figure 4: Ablation studies for temperature, lambda, and noise magnitude effects. The x-axis represents the experimental conditions, while the y-axis shows the performance metric.

To assess this problem in the optics of misclassification detectors, we calibrated the soft-probabilities of the models with a temperature parameter [7]. Note that this temperature has not necessarily the same value as the detection hyperparameter temperature. This calibration method is simple and effective, achieving performance close to state-of-the-art [14]. To measure how calibrated the model is before and after temperature scaling, we measured the expected calibration error (ECE) [7] before, with $T = 1$, and after calibration. We obtained the optimal temperature after a cross-validation procedure on the tuning set and measured the detection performance of the detection methods over the calibrated model on the test set. For the detection methods, we use the optimal temperature obtained from calibration, and no input pre-processing is conducted ($\epsilon = 0$), to observe precisely what is the effect of calibration. We set $\lambda = 0.5$.

Table 4 shows the detection performance over the calibrated models. We cannot conclude much from the CIFAR benchmark as the models are already well calibrated out of the training, with ECE of around 0.03. In general, calibrating the models slightly improved performance on this benchmark. However, for the ImageNet benchmark, we observe that Doctor gained a lot from the calibration, while REL-U remained more or less invariant to calibration on ImageNet. This implies that the performance of REL-U are robust under model's calibration.

Table 4: Impact of model probability calibration on misclassification detection methods. The uncalibrated and the calibrated performances are in terms of average FPR at 95% TPR (lower is better) and one standard deviation in parenthesis.

Architecture	Dataset	ECE_1	ECE_T	Uncal. Doctor	Cal. Doctor	Uncal. REL-U	Cal. REL-U
DenseNet-121	CIFAR-10	0.03	0.01	31.1 (2.4)	28.2 (3.8)	32.7 (1.7)	27.7 (2.1)
	CIFAR-100	0.03	0.01	44.4 (1.1)	45.9 (0.9)	45.7 (0.9)	46.6 (0.6)
ResNet-34	CIFAR-10	0.03	0.01	24.3 (0.0)	23.0 (1.4)	26.2 (0.0)	24.2 (0.1)
	CIFAR-100	0.06	0.04	40.0 (0.3)	38.7 (1.0)	40.6 (0.7)	38.9 (0.9)
ResNet-50	ImageNet	0.41	0.03	76.0 (0.0)	55.4 (0.7)	51.7 (0.0)	53.0 (0.3)

A.5 Mismatched Data

So far, we have evaluated methods for misclassification detection under the assumption that the data available to learn the uncertainty measure and that during testing are drawn from the same distribution.

In this section, we consider cases in which this assumption does not hold true, leading to a mismatch between the generative distributions of the data. Specifically, we investigate two sources of mismatch: *i)* Datasets with different label domains, where the symbol sets and symbols cardinality are different in each dataset; *ii)* Perturbation of the feature space domain generated using popular distortion filters. Understanding how machine learning models and misclassification detectors perform under such conditions can help us gauge and evaluate their robustness.

Mismatch from different label domains. We considered pre-trained classifiers on the CIFAR-10 dataset and evaluated their performance on detecting samples in CIFAR-10 and distinguishing them from samples in CIFAR-100, which has a different label domain. Similar experiments have been conducted in [17, 4, 23]. To explore the impact of dataset splits on machine learning models, Samples used for training were not reused for validation or evaluation. The test splits were divided into a validation set and an evaluation set, with the validation set consisting of 10%, 20%, 33%, or 50% of the total test split and samples used for training were not reused. In order to reduce the overlap between the label domain of CIFAR-10 and CIFAR-100, in this experimental setup we have ignored the samples corresponding to the following classes in CIFAR-100: bus, camel, cattle, fox, leopard, lion, pickup truck, streetcar, tank, tiger, tractor, train, and wolf.

For each split, we combine the number of validation samples from CIFAR-10 with an equal number of samples from CIFAR-100. In order to assess the validity of our results, each split has been randomly selected 10 times, and the results are reported in terms of mean and standard deviation in Figure 7. We observe how our proposed data-driven method performs when samples are provided to accurately describe the two groups. In order to reduce the overlap between the two datasets, and in line with previous work [4], we removed the classes in CIFAR-100 that most closely resemble the classes in CIFAR-10.

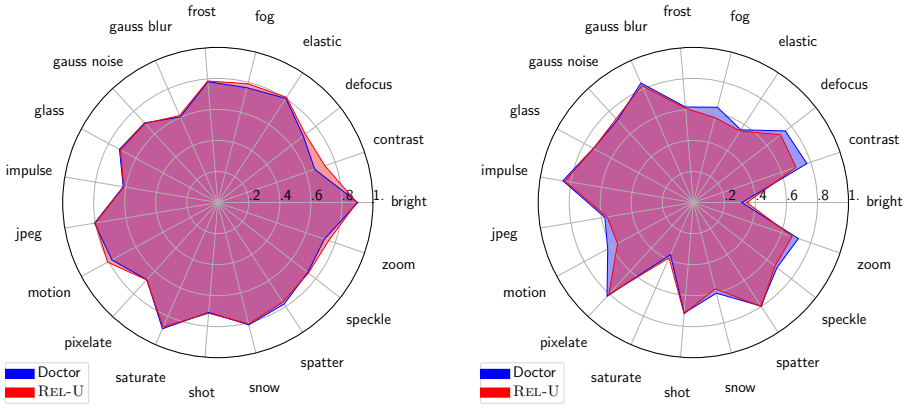
Mismatch from feature space corruption. We trained a model on the CIFAR-10 dataset and evaluated its ability to detect misclassification on the popular CIFAR-10C corrupted dataset, which contains a version of the classic CIFAR-10 test set perturbed according to 19 different types of corruptions and 5 levels of intensities. With this experiment we aim at investigating if our proposed detector is able to spot misclassifications that arise from input perturbation, based on the sole knowledge of the mislcassified patterns within the CIFAR-10 test split.

Table 5: We report the gap in accuracy between the original and the corrupted test set for the considered model. The gap is reported and average and standard deviation over the 19 different types of corruptions for corruption intensity equal to 5. The maximum and minimum gap are also reported, with the relative corruption type.

Architecture	Average gap	Max gap	Min gap
DenseNet121	0.36 ± 0.18	0.66 (Gaussian Blur)	0.04 (Brightness)
ResNet34	0.35 ± 0.20	0.72 (Impulse Noise)	0.03 (Brightness)

Consistent with previous experiments, we ensure that no samples from the training split are reused during validation and evaluation. To explore the effect of varying split sizes, we divide the test splits into validation and evaluation sets, with validation sets consisting of 10%, 20%, 33%, or 50% of the total test split. Each split has been produced 10 times with 10 different seeds and the average of the results has been reported in the radar plots in Figures 5 and 8. In the case of datasets with perturbed feature spaces, we solely utilize information from the validation samples in CIFAR-10 to detect misclassifications in the perturbed instances of the evaluation datasets, without using corrupted data during validation. We present visual plots that demonstrate the superior performance achieved by our proposed method compared to other methods. Additionally, for the case of perturbed feature spaces, we introduce radar plots, in which each vertex corresponds to a specific perturbation type, and report results for intensity 5. This particular choice of intensity is motivated by the fact that it creates the most relevant divergence between the accuracy of the model on the original test split and the accuracy of the model on the perturbed test split. Indeed the average gap in accuracy between the original test split and the perturbed test split is reported in Table 5.

We observe that our proposed method outperforms Doctor in terms of AUC and FPR, as demonstrated by the radar plots. As we can see, in the case of CIFAR-10 vs CIFAR-10C, the radar plots (Figures 5 and 8) show how the area covered by the AUC values achieves similar or larger values for the proposed



(a) AUC (b) FPR at 95% TPR

Figure 5: CIFAR-10 vs CIFAR-10C, DenseNet-121, using 10% of the test split for validation.

method, indeed confirming that it is able to better detect misclassifications in the mismatched data. Moreover, the FPR values are lower for the proposed method. For completeness, we report the error bar tables in Tables 6 and 7. Additionally, as a particular case of mismatch from feature space corruption, we have considered the task of detecting mismatch between MNIST and SVHN, the results are reported in Figure 6.

Table 6: DenseNet-121, error bar table, mismatch from different feature space corruption

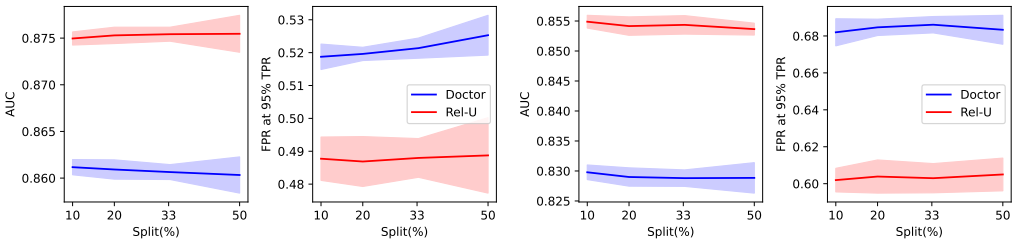
		Doctor		Rel-U	
Corruption	Split (%)	AUC	FPR	AUC	FPR
Brightness	10	0.90 ± 0.00	0.31 ± 0.00	0.90 ± 0.01	0.35 ± 0.03
	20	0.90 ± 0.00	0.31 ± 0.00	0.90 ± 0.00	0.32 ± 0.01
	33	0.90 ± 0.00	0.31 ± 0.00	0.90 ± 0.00	0.32 ± 0.01
	50	0.90 ± 0.00	0.31 ± 0.00	0.90 ± 0.00	0.32 ± 0.00
Contrast	10	0.66 ± 0.02	0.77 ± 0.03	0.73 ± 0.02	0.70 ± 0.02
	20	0.66 ± 0.02	0.77 ± 0.02	0.73 ± 0.01	0.69 ± 0.02
	33	0.67 ± 0.01	0.76 ± 0.01	0.74 ± 0.01	0.68 ± 0.01
	50	0.66 ± 0.01	0.77 ± 0.01	0.74 ± 0.01	0.67 ± 0.01
Defocus blur	10	0.70 ± 0.01	0.75 ± 0.00	0.72 ± 0.03	0.71 ± 0.05
	20	0.70 ± 0.01	0.75 ± 0.00	0.73 ± 0.01	0.69 ± 0.01
	33	0.70 ± 0.00	0.75 ± 0.00	0.73 ± 0.01	0.70 ± 0.01
	50	0.70 ± 0.00	0.75 ± 0.00	0.73 ± 0.01	0.71 ± 0.01
Elastic transform	10	0.80 ± 0.01	0.56 ± 0.00	0.81 ± 0.01	0.55 ± 0.02
	20	0.80 ± 0.01	0.56 ± 0.00	0.82 ± 0.00	0.53 ± 0.02
	33	0.80 ± 0.00	0.56 ± 0.00	0.82 ± 0.00	0.53 ± 0.01
	50	0.80 ± 0.00	0.56 ± 0.00	0.82 ± 0.00	0.53 ± 0.01
Fog	10	0.76 ± 0.01	0.63 ± 0.01	0.79 ± 0.01	0.56 ± 0.03
	20	0.76 ± 0.01	0.63 ± 0.01	0.79 ± 0.01	0.55 ± 0.02
	33	0.77 ± 0.00	0.63 ± 0.01	0.80 ± 0.00	0.56 ± 0.02
	50	0.77 ± 0.00	0.63 ± 0.00	0.80 ± 0.00	0.55 ± 0.01
Frost	10	0.78 ± 0.00	0.62 ± 0.00	0.79 ± 0.01	0.61 ± 0.02
	20	0.78 ± 0.00	0.62 ± 0.00	0.79 ± 0.01	0.59 ± 0.02
	33	0.78 ± 0.00	0.62 ± 0.00	0.80 ± 0.00	0.59 ± 0.01
	50	0.78 ± 0.00	0.62 ± 0.00	0.80 ± 0.00	0.59 ± 0.01
Gaussian blur	10	0.60 ± 0.00	0.84 ± 0.00	0.61 ± 0.05	0.82 ± 0.05
	20	0.60 ± 0.00	0.84 ± 0.00	0.63 ± 0.03	0.82 ± 0.02
	33	0.60 ± 0.00	0.84 ± 0.00	0.62 ± 0.02	0.82 ± 0.01

	50	0.60 ± 0.00	0.84 ± 0.00	0.61 ± 0.02	0.83 ± 0.01
Gaussian noise	10	0.70 ± 0.00	0.72 ± 0.00	0.69 ± 0.02	0.73 ± 0.02
	20	0.70 ± 0.00	0.72 ± 0.00	0.71 ± 0.01	0.72 ± 0.01
	33	0.70 ± 0.00	0.72 ± 0.00	0.70 ± 0.01	0.73 ± 0.01
	50	0.70 ± 0.00	0.72 ± 0.00	0.70 ± 0.01	0.73 ± 0.01
Glass blur	10	0.72 ± 0.00	0.73 ± 0.00	0.71 ± 0.01	0.73 ± 0.01
	20	0.72 ± 0.00	0.73 ± 0.00	0.72 ± 0.01	0.72 ± 0.01
	33	0.72 ± 0.00	0.73 ± 0.00	0.72 ± 0.01	0.73 ± 0.00
	50	0.72 ± 0.00	0.73 ± 0.00	0.72 ± 0.00	0.73 ± 0.00
Impulse noise	10	0.62 ± 0.00	0.85 ± 0.00	0.61 ± 0.03	0.84 ± 0.01
	20	0.62 ± 0.00	0.85 ± 0.00	0.63 ± 0.02	0.83 ± 0.01
	33	0.62 ± 0.00	0.85 ± 0.00	0.62 ± 0.01	0.84 ± 0.01
	50	0.62 ± 0.00	0.85 ± 0.00	0.62 ± 0.01	0.84 ± 0.01
Jpeg compression	10	0.81 ± 0.00	0.58 ± 0.00	0.80 ± 0.01	0.56 ± 0.02
	20	0.81 ± 0.00	0.58 ± 0.00	0.80 ± 0.00	0.55 ± 0.01
	33	0.81 ± 0.00	0.58 ± 0.00	0.81 ± 0.00	0.55 ± 0.01
	50	0.81 ± 0.00	0.58 ± 0.00	0.81 ± 0.00	0.55 ± 0.01
Motion blur	10	0.78 ± 0.01	0.63 ± 0.00	0.81 ± 0.01	0.56 ± 0.02
	20	0.78 ± 0.01	0.63 ± 0.00	0.82 ± 0.01	0.53 ± 0.02
	33	0.78 ± 0.00	0.63 ± 0.00	0.82 ± 0.00	0.54 ± 0.02
	50	0.78 ± 0.00	0.63 ± 0.00	0.82 ± 0.00	0.54 ± 0.01
Pixelate	10	0.68 ± 0.00	0.82 ± 0.00	0.68 ± 0.03	0.80 ± 0.01
	20	0.68 ± 0.00	0.82 ± 0.00	0.67 ± 0.03	0.81 ± 0.01
	33	0.68 ± 0.00	0.82 ± 0.00	0.66 ± 0.02	0.81 ± 0.01
	50	0.68 ± 0.00	0.82 ± 0.00	0.67 ± 0.02	0.81 ± 0.01
Saturate	10	0.89 ± 0.00	0.37 ± 0.01	0.88 ± 0.01	0.39 ± 0.03
	20	0.89 ± 0.00	0.37 ± 0.01	0.88 ± 0.00	0.36 ± 0.01
	33	0.89 ± 0.00	0.37 ± 0.00	0.88 ± 0.00	0.37 ± 0.01
	50	0.89 ± 0.00	0.37 ± 0.00	0.88 ± 0.00	0.36 ± 0.01
Shot noise	10	0.71 ± 0.00	0.72 ± 0.00	0.72 ± 0.02	0.72 ± 0.02
	20	0.71 ± 0.00	0.72 ± 0.00	0.73 ± 0.01	0.70 ± 0.02
	33	0.71 ± 0.00	0.72 ± 0.00	0.73 ± 0.01	0.70 ± 0.01
	50	0.71 ± 0.00	0.72 ± 0.00	0.73 ± 0.01	0.71 ± 0.01
Snow	10	0.81 ± 0.00	0.60 ± 0.00	0.81 ± 0.01	0.57 ± 0.01
	20	0.81 ± 0.00	0.60 ± 0.00	0.81 ± 0.01	0.57 ± 0.02
	33	0.81 ± 0.00	0.60 ± 0.00	0.81 ± 0.00	0.57 ± 0.01
	50	0.81 ± 0.00	0.60 ± 0.00	0.81 ± 0.00	0.57 ± 0.00
Spatter	10	0.78 ± 0.00	0.80 ± 0.00	0.77 ± 0.02	0.80 ± 0.04
	20	0.78 ± 0.00	0.80 ± 0.00	0.77 ± 0.01	0.79 ± 0.03
	33	0.78 ± 0.00	0.80 ± 0.00	0.77 ± 0.01	0.80 ± 0.02
	50	0.78 ± 0.00	0.80 ± 0.00	0.77 ± 0.00	0.80 ± 0.02
Speckle noise	10	0.73 ± 0.00	0.68 ± 0.00	0.74 ± 0.02	0.67 ± 0.03
	20	0.73 ± 0.00	0.68 ± 0.00	0.75 ± 0.01	0.65 ± 0.02
	33	0.73 ± 0.00	0.68 ± 0.00	0.75 ± 0.01	0.65 ± 0.01
	50	0.73 ± 0.00	0.68 ± 0.00	0.75 ± 0.01	0.66 ± 0.01
Zoom blur	10	0.73 ± 0.01	0.72 ± 0.01	0.76 ± 0.01	0.67 ± 0.04
	20	0.73 ± 0.01	0.71 ± 0.00	0.76 ± 0.01	0.65 ± 0.02
	33	0.73 ± 0.00	0.72 ± 0.00	0.77 ± 0.01	0.66 ± 0.02
	50	0.73 ± 0.00	0.72 ± 0.00	0.77 ± 0.01	0.67 ± 0.01

Table 7: ResNet-34, error bar table, mismatch from different feature space corruption

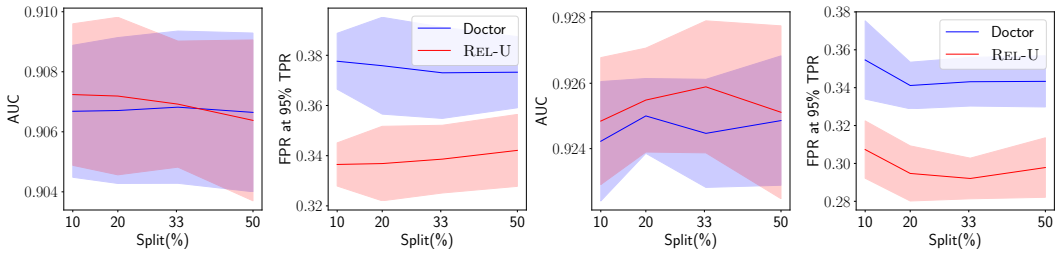
		Doctor		REL-U	
Corruption	Split (%)	AUC	FPR	AUC	FPR
Brightness	10	0.91 ± 0.00	0.30 ± 0.02	0.91 ± 0.01	0.33 ± 0.06
	20	0.91 ± 0.00	0.30 ± 0.01	0.92 ± 0.00	0.30 ± 0.02
	33	0.91 ± 0.00	0.30 ± 0.01	0.92 ± 0.00	0.30 ± 0.01
	50	0.92 ± 0.00	0.30 ± 0.01	0.92 ± 0.00	0.31 ± 0.01
Contrast	10	0.66 ± 0.03	0.76 ± 0.03	0.70 ± 0.02	0.68 ± 0.03
	20	0.66 ± 0.02	0.76 ± 0.03	0.71 ± 0.01	0.67 ± 0.02
	33	0.66 ± 0.02	0.75 ± 0.02	0.72 ± 0.01	0.66 ± 0.02
	50	0.66 ± 0.01	0.75 ± 0.01	0.72 ± 0.01	0.66 ± 0.01
Defocus blur	10	0.75 ± 0.02	0.60 ± 0.01	0.82 ± 0.01	0.49 ± 0.01
	20	0.75 ± 0.01	0.60 ± 0.01	0.82 ± 0.01	0.49 ± 0.01
	33	0.76 ± 0.01	0.60 ± 0.00	0.82 ± 0.00	0.50 ± 0.01
	50	0.76 ± 0.01	0.60 ± 0.00	0.82 ± 0.00	0.50 ± 0.01
Elastic transform	10	0.81 ± 0.02	0.53 ± 0.01	0.84 ± 0.01	0.45 ± 0.01
	20	0.81 ± 0.01	0.52 ± 0.01	0.85 ± 0.00	0.44 ± 0.01
	33	0.81 ± 0.01	0.52 ± 0.00	0.85 ± 0.00	0.44 ± 0.01
	50	0.81 ± 0.01	0.52 ± 0.00	0.85 ± 0.00	0.44 ± 0.00
Fog	10	0.73 ± 0.02	0.78 ± 0.05	0.81 ± 0.01	0.56 ± 0.02
	20	0.73 ± 0.01	0.77 ± 0.03	0.81 ± 0.01	0.57 ± 0.03
	33	0.74 ± 0.01	0.77 ± 0.03	0.81 ± 0.01	0.59 ± 0.02
	50	0.74 ± 0.01	0.77 ± 0.02	0.82 ± 0.00	0.59 ± 0.03
Frost	10	0.80 ± 0.00	0.65 ± 0.02	0.81 ± 0.01	0.60 ± 0.05
	20	0.80 ± 0.00	0.65 ± 0.01	0.82 ± 0.00	0.59 ± 0.02
	33	0.80 ± 0.00	0.65 ± 0.01	0.82 ± 0.00	0.59 ± 0.01
	50	0.80 ± 0.00	0.65 ± 0.01	0.82 ± 0.00	0.58 ± 0.01
Gaussian blur	10	0.71 ± 0.01	0.72 ± 0.00	0.75 ± 0.01	0.65 ± 0.01
	20	0.71 ± 0.00	0.72 ± 0.00	0.75 ± 0.01	0.66 ± 0.01
	33	0.71 ± 0.00	0.72 ± 0.00	0.75 ± 0.00	0.66 ± 0.01
	50	0.71 ± 0.00	0.72 ± 0.00	0.75 ± 0.00	0.67 ± 0.01
Gaussian noise	10	0.60 ± 0.00	0.85 ± 0.01	0.60 ± 0.03	0.87 ± 0.02
	20	0.60 ± 0.00	0.85 ± 0.01	0.61 ± 0.01	0.87 ± 0.01
	33	0.60 ± 0.00	0.85 ± 0.00	0.61 ± 0.01	0.87 ± 0.01
	50	0.60 ± 0.00	0.85 ± 0.00	0.61 ± 0.01	0.87 ± 0.00
Glass blur	10	0.72 ± 0.00	0.72 ± 0.00	0.73 ± 0.01	0.70 ± 0.03
	20	0.72 ± 0.00	0.72 ± 0.00	0.74 ± 0.01	0.69 ± 0.01
	33	0.72 ± 0.00	0.72 ± 0.00	0.74 ± 0.00	0.70 ± 0.01
	50	0.72 ± 0.00	0.71 ± 0.00	0.74 ± 0.00	0.69 ± 0.00
Impulse noise	10	0.63 ± 0.00	0.82 ± 0.00	0.66 ± 0.02	0.80 ± 0.03
	20	0.63 ± 0.00	0.82 ± 0.00	0.66 ± 0.01	0.80 ± 0.01
	33	0.63 ± 0.00	0.82 ± 0.00	0.66 ± 0.01	0.80 ± 0.01
	50	0.63 ± 0.00	0.82 ± 0.00	0.67 ± 0.01	0.80 ± 0.00
Jpeg compression	10	0.81 ± 0.01	0.57 ± 0.02	0.82 ± 0.01	0.51 ± 0.03
	20	0.81 ± 0.01	0.56 ± 0.01	0.83 ± 0.00	0.50 ± 0.01
	33	0.81 ± 0.00	0.57 ± 0.01	0.83 ± 0.00	0.51 ± 0.01
	50	0.81 ± 0.00	0.57 ± 0.00	0.83 ± 0.00	0.51 ± 0.01
Motion blur	10	0.78 ± 0.01	0.59 ± 0.02	0.83 ± 0.01	0.47 ± 0.01
	20	0.78 ± 0.01	0.58 ± 0.01	0.84 ± 0.01	0.47 ± 0.01
	33	0.78 ± 0.01	0.58 ± 0.01	0.84 ± 0.00	0.48 ± 0.01
	50	0.78 ± 0.00	0.57 ± 0.00	0.84 ± 0.00	0.48 ± 0.01
Pixelate	10	0.73 ± 0.00	0.70 ± 0.01	0.73 ± 0.02	0.69 ± 0.04
	20	0.73 ± 0.00	0.70 ± 0.01	0.74 ± 0.02	0.69 ± 0.03

	33	0.73 ± 0.00	0.70 ± 0.01	0.74 ± 0.01	0.69 ± 0.02
	50	0.73 ± 0.00	0.70 ± 0.00	0.74 ± 0.01	0.68 ± 0.01
Saturate	10	0.90 ± 0.00	0.31 ± 0.01	0.90 ± 0.01	0.32 ± 0.08
	20	0.90 ± 0.00	0.31 ± 0.00	0.91 ± 0.00	0.30 ± 0.01
	33	0.90 ± 0.00	0.31 ± 0.00	0.91 ± 0.00	0.30 ± 0.01
	50	0.90 ± 0.00	0.31 ± 0.00	0.91 ± 0.00	0.29 ± 0.01
Shot noise	10	0.63 ± 0.00	0.86 ± 0.01	0.65 ± 0.03	0.86 ± 0.04
	20	0.63 ± 0.00	0.85 ± 0.01	0.65 ± 0.01	0.86 ± 0.01
	33	0.63 ± 0.00	0.86 ± 0.00	0.65 ± 0.01	0.86 ± 0.02
	50	0.63 ± 0.00	0.86 ± 0.00	0.65 ± 0.01	0.86 ± 0.00
Snow	10	0.84 ± 0.00	0.55 ± 0.03	0.85 ± 0.01	0.49 ± 0.03
	20	0.84 ± 0.00	0.55 ± 0.02	0.85 ± 0.00	0.48 ± 0.02
	33	0.84 ± 0.00	0.55 ± 0.01	0.85 ± 0.00	0.48 ± 0.02
	50	0.84 ± 0.00	0.56 ± 0.01	0.85 ± 0.00	0.48 ± 0.01
Spatter	10	0.83 ± 0.00	0.59 ± 0.02	0.82 ± 0.01	0.60 ± 0.06
	20	0.83 ± 0.00	0.58 ± 0.01	0.83 ± 0.01	0.58 ± 0.04
	33	0.83 ± 0.00	0.59 ± 0.01	0.83 ± 0.01	0.58 ± 0.02
	50	0.83 ± 0.00	0.59 ± 0.00	0.83 ± 0.00	0.58 ± 0.01
Speckle noise	10	0.68 ± 0.00	0.81 ± 0.01	0.70 ± 0.03	0.79 ± 0.06
	20	0.68 ± 0.00	0.81 ± 0.01	0.70 ± 0.01	0.78 ± 0.03
	33	0.68 ± 0.00	0.81 ± 0.00	0.70 ± 0.01	0.79 ± 0.02
	50	0.68 ± 0.00	0.81 ± 0.00	0.70 ± 0.01	0.79 ± 0.01
Zoom blur	10	0.79 ± 0.01	0.58 ± 0.01	0.84 ± 0.01	0.47 ± 0.02
	20	0.79 ± 0.01	0.58 ± 0.00	0.84 ± 0.00	0.48 ± 0.01
	33	0.79 ± 0.01	0.58 ± 0.00	0.84 ± 0.00	0.49 ± 0.01
	50	0.79 ± 0.00	0.58 ± 0.00	0.84 ± 0.00	0.49 ± 0.01



(a) DenseNet-121. (b) ResNet-34.

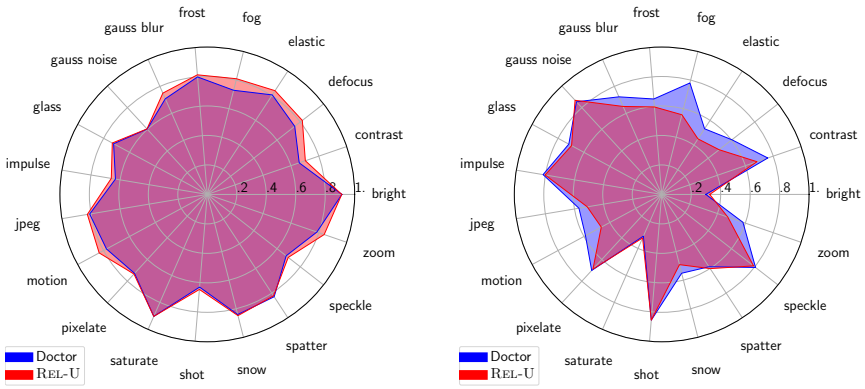
Figure 6: SVHN versus MNIST mismatch analysis.



(a) DenseNet-121 (b) ResNet-34

Figure 7: Impact of different validation set sizes (in percentage of test split) for mismatch detection.

Tables 6 and 7 report results for multiple splits, other than those reported in Figures 5 and 8.



(a) AUC

(b) FPR at 95% TPR

Figure 8: CIFAR-10 vs CIFAR-10C, ResNet-34, using 10% of the test split for validation.