On selective classification under distribution shift

Luís Felipe P. Cattelan, Danilo Silva
Machine Learning and Applications Research Group
Federal University of Santa Catarina
Florianópolis - SC, Brazil
lfp.cattelan@gmail.com, danilo.silva@ufsc.br

Abstract

This paper addresses the problem of selective classification for deep neural networks, where a model is allowed to abstain from low-confidence predictions to avoid potential errors. Specifically, we investigate whether the selective classification performance of ImageNet classifiers is robust to distribution shift. Motivated by the intriguing observation in recent work that many classifiers appear to have a "broken" confidence estimator, we start by evaluating methods to fix this issue. We focus on so-called post-hoc methods, which replace the confidence estimator of a given classifier without retraining or modifying it, thus being practically appealing. We perform an extensive experimental study of many existing and proposed confidence estimators applied to 84 pre-trained ImageNet classifiers available from popular repositories. Our results show that a simple p-norm normalization of the logits, followed by taking the maximum logit as the confidence estimator, can lead to considerable gains in selective classification performance, completely fixing the pathological behavior observed in many classifiers. As a consequence, the selective classification performance of any classifier becomes almost entirely determined by its corresponding accuracy. Then, we show these results are consistent under distribution shift: a method that enhances performance in the in-distribution scenario also provides similar gains under distribution shift. Moreover, although a slight degradation in selective classification performance is observed under distribution shift, this can be explained by the drop in accuracy of the classifier, together with the slight dependence of selective classification performance on accuracy.

1 Introduction

A reliable predictive model must be able to identify cases where it is likely to make an incorrect prediction and withhold the output to prevent a wrong decision, i.e., it must be able to say "I don't know". This ability is essential in many real-world applications, such as medical diagnosis and autonomous driving, where the consequences of erroneous decisions can be severe [Zou et al., 2023, Neumann et al., 2018]. The general framework of rejecting predictions for which a classifier is least confident, hoping to increase the performance on the accepted predictions, is known as *selective classification* Geifman and El-Yaniv [2017], El-Yaniv and Wiener [2010].

A scenario where selective classification may be particularly relevant is the case of distribution shift, where the data seen during inference are taken from a distribution different from the training one, and the performance of a classifier is known to significantly drop [Recht et al., 2019a]. Previous work has shown that the calibration performance is also significantly degraded in these conditions, especially when post-hoc methods are applied [Ovadia et al., 2019a]. Thus, motivated by recent work that shows a trade-off between calibration and selective classification performance [Zhu et al., 2022], we wonder if the latter is also sensitive to distribution shift.

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

We start by investigating whether simple post-hoc methods can enhance the selective classification performance of deep neural networks with softmax outputs. We perform an extensive experimental study of many existing and proposed confidence estimators applied to 84 pre-trained ImageNet classifiers available from popular repositories. Our results show that a simple p-norm normalization of the logits, followed by taking the maximum logit as the confidence estimator, can lead to considerable gains in selective classification performance. Indeed, this approach fixes the pathological behavior observed by Galil et al. [2023], where many state-of-the-art ImageNet classifiers, despite attaining excellent predictive performance, nevertheless exhibit appallingly poor performance at detecting their own mistakes. We show that this pathology is a consequence of the confidence estimator used (the maximum softmax probability), not necessarily an innate limitation of the model. Therefore, when post-hoc optimization is applied, the selective classification performance of any classifier becomes almost entirely determined by its corresponding accuracy.

Then, we show these results are consistent under distribution shift: a method that enhances performance in the in-distribution scenario also provides similar gains under distribution shift, even when optimized on the former. Moreover, although a slight degradation in selective classification performance is observed under distribution shift, this can be explained by the drop in accuracy of the classifier, together with the slight dependence of selective classification performance on accuracy.

2 Background

2.1 Selective classification

Let P be an unknown distribution over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space, $\mathcal{Y} = \{1, \ldots, C\}$ is the label space, and C is the number of classes. The *risk* of a *classifier* $h : \mathcal{X} \to \mathcal{Y}$ is $R(h) = E_P[\ell(h(x), y)]$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ is a loss function, for instance, the 0/1 loss $\ell(\hat{y}, y) = \mathbb{1}[\hat{y} \neq y]$, where $\mathbb{1}[\cdot]$ denotes the indicator function. A *selective classifier* [Geifman and El-Yaniv, 2017] is a pair (h, g), where h is a classifier and $g : \mathcal{X} \to \mathbb{R}$ is a *confidence estimator*, which quantifies the model's confidence on its prediction for a given input. For some fixed threshold t, given an input x, the selective model makes a prediction $h(x)$ if $g(x) \geq t$, otherwise the prediction is rejected. A selective model's *coverage* $\phi(h, g) = P[g(x) \geq t]$ is the probability mass of the selected samples in $\mathcal{X}$, while its *selective risk* $R(h, g) = E_P[\ell(h(x), y) \mid g(x) \geq t]$ is its risk restricted to the selected samples. In particular, a model's risk equals its selective risk at*full coverage* (i.e., for t such that $\phi(h, g) = 1$). These quantities can be evaluated empirically given a given a test dataset $\{(x_i, y_i)\}_{i=1}^N$ drawn i.i.d. from P, yielding the *empirical coverage* $\hat{\phi}(h, g) = (1/N) \sum_{i=1}^N \mathbb{1}[g(x_i) \geq t]$ and the *empirical selective risk*

$$\hat{R}(h, g) = \frac{\sum_{i=1}^N \ell(h(x_i), y_i) \mathbb{1}[g(x_i) \geq t]}{\sum_{i=1}^N \mathbb{1}[g(x_i) \geq t]}. \tag{1}$$

Note that, by varying t, it is generally possible to trade off coverage for selective risk, i.e., a lower selective risk can usually (but not necessarily always) be achieved if more samples are rejected. This tradeoff is captured by the *risk-coverage (RC) curve* [Geifman and El-Yaniv, 2017], a plot of $\hat{R}(h, g)$ as a function of $\hat{\phi}(h, g)$. (An example can be seen in Figure 4 in the Appendix.) While the RC curve provides a full picture of the performance of a selective classifier, it is convenient to have a scalar metric that summarizes this curve. A commonly used metric is the *area under the RC curve* (AURC) [Ding et al., 2020, Geifman et al., 2019], denoted by $\text{AURC}(h, g)$. Another interesting metric, which forces the choice of an operating point, is the *selective accuracy constraint* (SAC) [Galil et al., 2023], defined as the maximum coverage allowed for a model to achieve a specified accuracy.

2.2 Confidence estimation

We restrict attention to classifiers that can be decomposed as $h(x) = \hat{y} = \arg\max_{k \in \mathcal{Y}} z_k$, where $\mathbf{z} = f(x)$ and $f : \mathcal{X} \to \mathbb{R}^C$ is a neural network. The network output $\mathbf{z}$ is referred to as the (vector of) *logits* or *logit vector*, due to the fact that it is typically applied to a softmax function to obtain an estimate of the posterior distribution $P[y|x]$. The softmax function is defined as $\sigma : \mathbb{R}^C \to [0, 1]^C$, $\sigma_k(\mathbf{z}) = e^{z_k} / \sum_{j=1}^C e^{z_j}$, $k \in \{1, \ldots, C\}$, where $\sigma_k(\mathbf{z})$ denotes the kth element of the vector $\sigma(\mathbf{z})$.

The most popular confidence estimator, widely used as a baseline for selective classification and misclassification detection, is arguably the *maximum softmax probability* (MSP) [Ding et al., 2020]:

$$g(x) = \text{MSP}(\mathbf{z}) \triangleq \max_{k \in \mathcal{Y}} \sigma_k(\mathbf{z}) = \sigma_{\hat{y}}(\mathbf{z}) \tag{2}$$

Further examples of confidence estimators that can be computed from the logits are SoftmaxMargin, MaxLogit, LogitsMargin, NegativeEntropy and NegativeGini, defined in Appendix B.

3 Normalized AURC

A common criticism of the AURC metric is that it does not allow for meaningful comparisons across problems [Geifman et al., 2019]. An AURC of some arbitrary value, for instance, 0.05, may correspond to an ideal confidence estimator for one classifier (of much higher risk) and to a completely random confidence estimator for another classifier (of risk equal to 0.05). The excess AURC (E-AURC) was proposed by Geifman et al. [2019] to alleviate this problem: for a given classifier h and confidence estimator g, it is defined as E-AURC$(h, g) =$ AURC$(h, g) -$ AURC(h, g^*), where g^* corresponds to a hypothetically optimal confidence estimator that perfectly orders samples in decreasing order of their losses. Thus, an ideal confidence estimator always has zero E-AURC.

Unfortunately, E-AURC is still highly sensitive to the classifier's risk, as shown by Galil et al. [2023], who suggested the use of AUROC instead. However, using AUROC for comparing confidence estimators has an intrinsic disadvantage: if we are using AUROC to evaluate the performance of a tunable confidence estimator, it makes sense to optimize it using this same metric. However, as AUROC and AURC are not necessarily monotonically aligned Ding et al. [2020], the resulting confidence estimator will be optimized for a different problem than the one in which we were originally interested (which is selective classification). Ideally, we would like to evaluate confidence estimators using a metric that is a monotonic function of AURC.

We propose a simple modification to E-AURC that eliminates the shortcomings pointed out in [Galil et al., 2023]: normalizing by the E-AURC of a random confidence estimator, whose AURC is equal to the classifier's risk. More precisely, we define the normalized AURC (NAURC) as

$$\text{NAURC}(h, g) = \frac{\text{AURC}(h, g) - \text{AURC}(h, g^*)}{R(h) - \text{AURC}(h, g^*)}. \tag{3}$$

Note that this corresponds to a min-max scaling that maps the AURC of the ideal classifier to 0 and the AURC of the random classifier to 1. The resulting NAURC is suitable for comparison across different classifiers and is monotonically related to AURC.

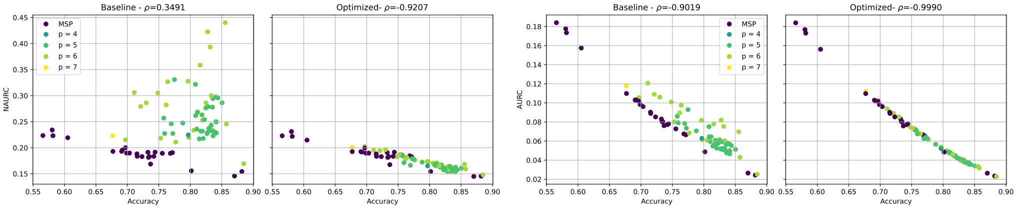
4 Post-hoc optimization of confidence estimators

4.1 Tunable Logit Transformations

We propose a simple but powerful framework for designing post-hoc confidence estimators: the idea is to take any parameter-free logit-based confidence estimator, such as those described in Section 2.2 and Appendix B, and augment it with a logit transformation parameterized by a few hyperparameters, which are then tuned (e.g., via grid search) using a labeled hold-out dataset (i.e., validation data). Moreover, this hyperparameter tuning is done using as objective function the AURC directly.

Here we consider two types of logit transformations: the first is *temperature scaling* (TS) [Guo et al., 2017], which consists in transforming the logits as $\mathbf{z}' = \mathbf{z}/T$, where $T > 0$ is called the temperature. The application of TS, where T is optimized using AURC, is denoted by the suffix -TS-AURC.

The second transformation is *logit p-normalization*, defined as the operation $\mathbf{z}' = \mathbf{z}/(\tau \|\mathbf{z}\|_p)$, where $\|\mathbf{z}\|_p \triangleq (|z_1|^p + \cdots + |z_C|^p)^{1/p}$, $p \in \mathbb{R}$, is the p-norm of $\mathbf{z}$ and $\tau > 0$ is a TS parameter. This is inspired by Wei et al. [2022], who proposed logit 2-normalization during training to improve calibration and out-of-distribution detection. Applying logit p-normalization, together with the optimization of p and τ using AURC, is referred to here by the suffix -pNorm. Note that, different from the cross entropy loss used in the standard TS, the presented selective classification metrics are not differentiable. Nevertheless, since we are only dealing with 2 parameters, a grid-search method can be applied. Also, we note that a relative small range of values need to be tested—typically, the



(a) NAURC

(b) AURC

Figure 1: NAURC and AURC of 84 ImageNet classifiers with respect to their accuracy, before and after post-hoc optimization with MaxLogit-pNorm. ρ is the Spearman correlation between the metric and the corresponding accuracy. The legend shows the optimal value of p for each model, where MSP indicates MSP fallback (no significant positive gain).

optimal temperature with respect to selective classification is between 0 and 1 [Galil et al., 2023], and the p-norm tend to converge quickly to the maximum absolute value of the vector as p increases. In our experiments, we noticed that it suffices to evaluate a few integer values of p and temperature values between 0.01 and 3 with a step size of 0.01. More details on this optimization can be found in Appendix E.

4.2 Choosing the best confidence estimator

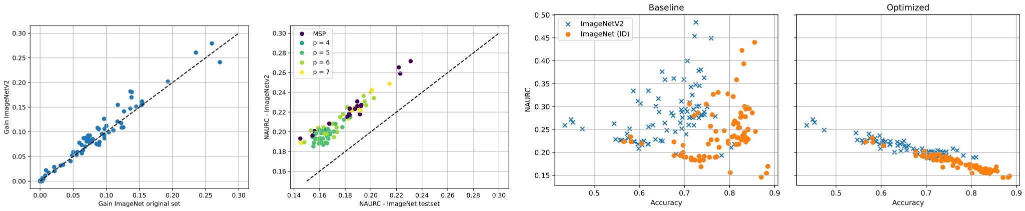
In Galil et al. [2023], the authors show that some ImageNet classifiers are innately better than others in misclassification detection. However, the authors benchmark the models using the MSP exclusively as the selective mechanism. Here we follow an approach analogous to the ones advocated in Wang et al. [2021] and Ashukha et al. [2020] in the context of calibration: when comparing different models, we should use the best known confidence estimator for each of them, including post-hoc optimization.

Thus, we start by performing an extensive evaluation of confidence estimators. In Appendix D experiments are conducted using 84 pre-trained ImageNet classifiers available from popular repositories. Every possible combination of a confidence estimator listed in Section 2.2 with a logit transformation described in Section (TS-AURC, pNorm or none) is evaluated to determine the best selective mechanism for each model. Our results show that the phenomenon reported by Galil et al. [2023] is not really about the capacity of models in estimating uncertainty, but mainly due to the fact that, for some models, the MSP is a "broken" confidence estimator. For these models, other confidence estimation methods can be used to attain considerably better performance—in particular, MaxLogit-pNorm is consistently observed to be the best choice, especially when data efficiency is taken into account (see Appendix G). However, for models for which the MSP is already a good confidence estimator, these alternative functions can even degrade the performance, in which case a fallback to the MSP is taken (see Appendix C).

Post-hoc optimization fixes the anomalies reported by Galil et al. [2023], which can be seen in two ways: in Figure 1a, after optimization, all models exhibit a similar level of confidence estimation performance (as measured by NAURC), although we can still see some dependency on accuracy (better predictive models are slightly better at predicting their own failures). In Figure 1b, it is clear that, after optimization, the selective classification performance of any classifier (as measured by AURC) becomes almost entirely determined by its corresponding accuracy. Indeed, the Spearman correlation between AURC and accuracy becomes extremely close to 1, which implies that any "broken" confidence estimators have been fixed and, consequently, total accuracy becomes the primary determinant of selective performance even at lower coverage levels.

5 Selective classification robustness to distribution shift

We now turn to the question of how post-hoc methods for selective classification perform under distribution shift. Previous works have shown that calibration can be harmed under distribution shift, especially when certain post-hoc methods—such as temperature scaling—are applied [Ovadia et al., 2019b]. To investigate whether a similar issue occurs for selective classification, we evaluate selected post-hoc methods on ImageNet-C [Hendrycks and Dietterich, 2018], which consists in 15 different



(a) (b) (c)

Figure 2: (a) NAURC gains (over MSP) on ImageNetV2 versus NAURC gains on the ImageNet test set. (b) NAURC on ImageNetV2 versus NAURC on the ImageNet test set. (c) NAURC versus accuracy. All models are optimized using MaxLogit-pNorm.

Table 1: Selective classification performance for a ResNet-50 on ImageNet under distribution shift. For ImageNet-C, each entry is the average across all corruption types for a given level of corruption. The target accuracy is the one achieved for corruption level 0 (i.e., 80.86%).

		Corruption level						
	Method	0	1	2	3	4	5	V2
Accuracy [%]	-	80.86	68.56	60.03	51.85	39.44	27.09	69.97
Coverage (SAC) [%]	MSP	100	71.97	52.17	37.33	19.16	8.40	76.66
	MSP-TS-AURC	100	72.89	55.83	40.90	24.65	12.46	77.29
	MaxLogit-pNorm	100	**75.24**	**58.76**	**43.98**	**27.27**	**14.78**	**79.00**

corruptions of the ImageNet's validation set, and on ImageNetV2 [Recht et al., 2019b], which is an independent sampling of the ImageNet test set replicating the original dataset creation process. We follow the standard approach for evaluating robustness with these datasets, which is to use them only for inference; thus, the post-hoc methods are optimized using only the 5000 hold-out images from the uncorrupted ImageNet validation dataset, as discussed in Appendix D. To avoid data leakage, the same split is applied to the ImageNet-C dataset, so that inference is performed only on the 45000 images originally selected as the test set.

First, we evaluate the performance of MaxLogit-pNorm on ImageNet and ImageNetV2 for all classifiers considered. Figure 2a shows that the NAURC gains (over the MSP baseline) obtained for ImageNet translate to similar gains for ImageNetV2, showing that this post-hoc method is quite robust to distribution shift.

Then, considering all models after post-hoc optimization with MaxLogit-pNorm, we investigate whether selective classification performance itself (as measured by NAURC) is robust to distribution shift. As can be seen in Figure 2b, the results are consistent, following an affine function; however, a significant degradation in NAURC can be observed for all models under distribution shift. While at first sight this would suggest a lack of robustness, a closer look reveals that it can actually be explained by the natural accuracy drop of the underlying classifier under distribution shift. Indeed, we have already noticed in Figure 1a a negative correlation between the NAURC and the accuracy; in Figure 2c these results are expanded by including the evaluation on ImageNetV2, where we can see that the strong correlation between NAURC and accuracy continues to hold. Similar results are obtained when evaluating classifiers on ImageNet-C, as presented in Appendix H.

Finally, to give a more tangible illustration of the impact of selective classification, Table 1 shows the SAC metric for a ResNet50 under distribution shift, with the target accuracy as the original accuracy obtained with the in-distribution data. As can be seen, the original accuracy can be restored at the expense of coverage; meanwhile, MaxLogit-pNorm achieves higher coverages for all distribution shifts considered, significantly improving coverage over the MSP baseline.

References

M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi. A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges. *Information Fusion*, 76:243–297, Dec. 2021. ISSN 15662535. doi: 10.1016/j.inffus.2021.05.008. URL http://arxiv.org/abs/2011.06225. arXiv:2011.06225 [cs].

T. Abe, E. K. Buchanan, G. Pleiss, R. Zemel, and J. P. Cunningham. Deep Ensembles Work, But Are They Necessary? *Advances in Neural Information Processing Systems*, 35:33646–33660, Dec. 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/da18c47118a2d09926346f33bebde9f4-Abstract-Conference.html.

A. Ashukha, D. Molchanov, A. Lyzhov, and D. Vetrov. PITFALLS OF IN-DOMAIN UNCERTAINTY ESTIMATION AND ENSEMBLING IN DEEP LEARNING. 2020.

M. Ayhan and P. Berens. Test-time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks. Apr. 2018. URL https://www.semanticscholar.org/paper/Test-time-Data-Augmentation-for-Estimation-of-in-Ayhan-Berens/172df6d55b81f184ab0042c49634ccf9b72ed253.

S. A. Balanya, D. Ramos, and J. Maroñas. Adaptive Temperature Scaling for Robust Calibration of Deep Neural Networks, Mar. 2023. URL https://papers.ssrn.com/abstract=4379258.

M. I. Belghazi and D. Lopez-Paz. What classifiers know what they don't?, July 2021. URL http://arxiv.org/abs/2107.06217. arXiv:2107.06217 [cs].

L. F. P. Cattelan and D. Silva. On the performance of uncertainty estimation methods for deep-learning based image classification models. In *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, pages 532–543. SBC, Nov. 2022. doi: 10.5753/eniac.2022.227603. URL https://sol.sbc.org.br/index.php/eniac/article/view/22810. ISSN: 2763-9061.

L. Clarté, B. Loureiro, F. Krzakala, and L. Zdeborová. Expectation consistency for calibration of neural networks, Mar. 2023. URL http://arxiv.org/abs/2303.02644. arXiv:2303.02644 [cs, stat].

C. Corbière, N. Thome, A. Saporta, T.-H. Vu, M. Cord, and P. Pérez. Confidence Estimation via Auxiliary Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 6043–6055, Oct. 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3085983. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848. ISSN: 1063-6919.

Y. Ding, J. Liu, J. Xiong, and Y. Shi. Revisiting the Evaluation of Uncertainty Estimation and Its Application to Explore Model Complexity-Uncertainty Trade-Off. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 22–31, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72819-360-1. doi: 10.1109/CVPRW50498.2020.00010. URL https://ieeexplore.ieee.org/document/9150782/.

R. El-Yaniv and Y. Wiener. On the Foundations of Noise-free Selective Classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010. ISSN 1533-7928. URL http://jmlr.org/papers/v11/el-yaniv10a.html.

L. Feng, M. O. Ahmed, H. Hajimirsadeghi, and A. H. Abdi. Towards Better Selective Classification. Feb. 2023. URL https://openreview.net/forum?id=5gDz_yTcst.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR, June 2016. URL https://proceedings.mlr.press/v48/gal16.html. ISSN: 1938-7228.

I. Galil, M. Dabbah, and R. El-Yaniv. What Can we Learn From The Selective Prediction And Uncertainty Estimation Performance Of 523 Imagenet Classifiers? Feb. 2023. URL https://openreview.net/forum?id=p66AzKi6Xim&referrer=%5BAuthor%20Console%5D(%2Fgroup%3Fid%3DICLR.cc%2F2023%2FConference%2FAuthors%23your-submissions).

J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu. A Survey of Uncertainty in Deep Neural Networks, Jan. 2022. URL http://arxiv.org/abs/2107.03342. arXiv:2107.03342 [cs, stat].

Y. Geifman and R. El-Yaniv. Selective Classification for Deep Neural Networks, June 2017. URL http://arxiv.org/abs/1705.08500. arXiv:1705.08500 [cs].

Y. Geifman and R. El-Yaniv. SelectiveNet: A Deep Neural Network with an Integrated Reject Option. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2151–2159. PMLR, May 2019. URL https://proceedings.mlr.press/v97/geifman19a.html. ISSN: 2640-3498.

Y. Geifman, G. Uziel, and R. El-Yaniv. Bias-Reduced Uncertainty Estimation for Deep Neural Classifiers, Apr. 2019. URL http://arxiv.org/abs/1805.08206. arXiv:1805.08206 [cs, stat].

E. D. C. Gomes, M. Romanelli, F. Granese, and P. Piantanida. A simple Training-Free Method for Rejection Option. Sept. 2022. URL https://openreview.net/forum?id=K1DdnjL6p7.

J. Gonsior, C. Falkenberg, S. Magino, A. Reusch, M. Thiele, and W. Lehner. To Softmax, or not to Softmax: that is the question when applying Active Learning for Transformer Models, Oct. 2022. URL http://arxiv.org/abs/2210.03005. arXiv:2210.03005 [cs].

F. Granese, M. Romanelli, D. Gorla, C. Palamidessi, and P. Piantanida. Doctor: A simple method for detecting misclassification errors. *Advances in Neural Information Processing Systems*, 34: 5669–5681, 2021.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330. PMLR, July 2017. URL https://proceedings.mlr.press/v70/guo17a.html. ISSN: 2640-3498.

K. Hendrickx, L. Perini, D. V. der Plas, W. Meert, and J. Davis. Machine learning with a reject option: A survey. *ArXiv*, abs/2107.11277, 2021.

D. Hendrycks and T. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. Dec. 2018. URL https://openreview.net/forum?id=HJz6tiCqYm.

D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, J. Steinhardt, and D. Song. Scaling Out-of-Distribution Detection for Real-World Settings. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8759–8773. PMLR, June 2022. URL https://proceedings.mlr.press/v162/hendrycks22a.html.

L. Huang, C. Zhang, and H. Zhang. Self-Adaptive Training: beyond Empirical Risk Minimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 19365–19376. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e0ab531ec312161511493b002f9be2ee-Abstract.html.

Z. Jiang, J. Gu, and D. Z. Pan. NormSoftmax: Normalize the Input of Softmax to Accelerate and Stabilize Training. Feb. 2023. URL https://openreview.net/forum?id=4g7nCbpjNwd.

A. Karandikar, N. Cain, D. Tran, B. Lakshminarayanan, J. Shlens, M. C. Mozer, and R. Roelofs. Soft Calibration Objectives for Neural Networks. Nov. 2021. URL https://openreview.net/forum?id=-tVD13hOsQ3.

S. Kornblith, T. Chen, H. Lee, and M. Norouzi. Why Do Better Loss Functions Lead to Less Transferable Features? In *Advances in Neural Information Processing Systems*, volume 34, pages 28648–28662. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/hash/f0bf4a2da952528910047c31b6c2e951-Abstract.html`.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html`.

Y. Le Cun, O. Matan, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jacket, and H. Baird. Handwritten zip code recognition with multilayer networks. In *10th International Conference on Pattern Recognition [1990] Proceedings*, volume ii, pages 35–40 vol.2, June 1990. doi: 10.1109/ICPR.1990.119325.

L. Lebovitz, L. Cavigelli, M. Magno, and L. K. Muller. Efficient Inference With Model Cascades. *Transactions on Machine Learning Research*, May 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=obB415rg8q`.

S. Liang, Y. Li, and R. Srikant. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. May 2023. URL `https://openreview.net/forum?id=H1VGkIxRZ`.

W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Z. Liu, Z. Wang, P. P. Liang, R. R. Salakhutdinov, L.-P. Morency, and M. Ueda. Deep gamblers: Learning to abstain with portfolio theory. *Advances in Neural Information Processing Systems*, 32, 2019.

M. Lubrano, Y. Bellahsen-Harrar, R. Fick, C. Badoual, and T. Walter. Simple and efficient confidence score for grading whole slide images. *arXiv preprint arXiv:2303.04604*, 2023.

J. Moon, J. Kim, Y. Shin, and S. Hwang. Confidence-Aware Learning for Deep Neural Networks. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7034–7044. PMLR, Nov. 2020. URL `https://proceedings.mlr.press/v119/moon20a.html`. ISSN: 2640-3498.

J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. Torr, and P. Dokania. Calibrating Deep Neural Networks using Focal Loss. In *Advances in Neural Information Processing Systems*, volume 33, pages 15288–15299. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/aeb7b30ef1d024a76f21a1d40e30c302-Abstract.html`.

L. Neumann, A. Zisserman, and A. Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. 2018.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019a.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model' s uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b. URL `https://papers.nips.cc/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-Abstract.html`.

O. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Oxfordiiit pet dataset. [Online]. Available from: `https://www.robots.ox.ac.uk/~vgg/data/pets/`, 2012. Accessed: 2023-09-28.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance

Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

A. Rahimi, T. Mensink, K. Gupta, T. Ajanthan, C. Sminchisescu, and R. Hartley. Post-hoc Calibration of Neural Networks by g-Layers, Feb. 2022. URL http://arxiv.org/abs/2006.12807. arXiv:2006.12807 [cs, stat].

B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019a.

B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019b.

M. Shen, Y. Bu, P. Sattigeri, S. Ghosh, S. Das, and G. Wornell. Post-hoc Uncertainty Learning using a Dirichlet Meta-Model, Dec. 2022. URL http://arxiv.org/abs/2212.07359. arXiv:2212.07359 [cs].

M. Streeter. Approximation Algorithms for Cascading Prediction Models. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4752–4760. PMLR, July 2018. URL https://proceedings.mlr.press/v80/streeter18a.html.

M. Teye, H. Azizpour, and K. Smith. Bayesian Uncertainty Estimation for Batch Normalized Deep Networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4907–4916. PMLR, July 2018. URL https://proceedings.mlr.press/v80/teye18a.html. ISSN: 2640-3498.

C. Tomani, D. Cremers, and F. Buettner. Parameterized Temperature Scaling for Boosting the Expressive Power in Post-Hoc Uncertainty Calibration, Sept. 2022. URL http://arxiv.org/abs/2102.12182. arXiv:2102.12182 [cs].

D.-B. Wang, L. Feng, and M.-L. Zhang. Rethinking Calibration of Deep Neural Networks: Do Not Be Afraid of Overconfidence. In *Advances in Neural Information Processing Systems*, volume 34, pages 11809–11820. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/61f3a6dbc9120ea78ef75544826c814e-Abstract.html.

H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li. Mitigating Neural Network Overconfidence with Logit Normalization. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23631–23644. PMLR, June 2022. URL https://proceedings.mlr.press/v162/wei22d.html. ISSN: 2640-3498.

R. Wightman. Pytorch Image Model, 2019. URL https://github.com/huggingface/pytorch-image-models.

G. Xia and C.-S. Bouganis. On the Usefulness of Deep Ensemble Diversity for Out-of-Distribution Detection, Sept. 2022. URL http://arxiv.org/abs/2207.07517. arXiv:2207.07517 [cs].

X.-Y. Zhang, G.-S. Xie, X. Li, T. Mei, and C.-L. Liu. A Survey on Learning to Reject. *Proceedings of the IEEE*, 111(2):185–215, Feb. 2023. ISSN 1558-2256. doi: 10.1109/JPROC.2023.3238024. Conference Name: Proceedings of the IEEE.

F. Zhu, Z. Cheng, X.-Y. Zhang, and C.-L. Liu. Rethinking Confidence Calibration for Failure Prediction. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision – ECCV 2022*, volume 13685, pages 518–536. Springer Nature Switzerland, Cham, 2022. ISBN 978-3-031-19805-2 978-3-031-19806-9. doi: 10.1007/978-3-031-19806-9_30. URL https://link.springer.com/10.1007/978-3-031-19806-9_30. Series Title: Lecture Notes in Computer Science.

K. Zou, Z. Chen, X. Yuan, X. Shen, M. Wang, and H. Fu. A Review of Uncertainty Estimation and its Application in Medical Imaging, Feb. 2023. URL http://arxiv.org/abs/2302.08119. arXiv:2302.08119 [cs, eess].

Supplementary Material

A Related work

Selective prediction is also known as learning with a reject option (see [Zhang et al., 2023, Hendrickx et al., 2021] and references therein), where the rejector is usually a thresholded confidence estimator. Essentially the same problem is studied under the equivalent terms misclassification detection [Hendrycks and Gimpel, 2016], failure prediction [Corbière et al., 2022, Zhu et al., 2022], and (ordinal) ranking [Moon et al., 2020, Galil et al., 2023]. Uncertainty estimation is a more general term that encompasses these tasks (where confidence may be taken as negative uncertainty) as well as other tasks where uncertainty might be useful, such as calibration and out-of-distribution (OOD) detection, among others [Gawlikowski et al., 2022, Abdar et al., 2021]. These tasks are generally not aligned: for instance, optimizing for calibration may harm selective classification performance [Ding et al., 2020, Zhu et al., 2022, Galil et al., 2023]. Our focus here is on in-distribution selective classification, although we also study robustness to distribution shift.

Interestingly, the same principles of selective classification can be applied to enable efficient inference with model cascades [Lebovitz et al., 2023], although the literature on those topics appears disconnected.

Most approaches to selective classification consider the base model as part of the learning problem [Geifman and El-Yaniv, 2019, Huang et al., 2020, Liu et al., 2019], which we refer to as training-based approaches. While such an approach has a theoretical appeal, the fact that it requires retraining a model is a significant practical drawback. Alternatively, one may keep the model fixed and only modify or replace the confidence estimator, which is known as a post-hoc approach. Such an approach is practically appealing and perhaps more realistic, as it does not require retraining. Papers that follow this approach typically construct a *meta-model* that feeds on intermediate features of the base model and is trained to predict whether or not the base model is correct on hold-out samples [Corbière et al., 2022, Shen et al., 2022]. However, depending on the size of such a meta-model, its training may still be computationally demanding.

A popular tool in the uncertainty literature is the use of ensembles [Lakshminarayanan et al., 2017, Teye et al., 2018, Ayhan and Berens, 2018], of which Monte-Carlo dropout Gal and Ghahramani [2016] is a prominent example. While constructing a confidence estimator from ensemble component outputs may be considered post-hoc if the ensemble is already trained, the fact that multiple inference passes need to be performed significantly increases the computational burden at test time. Moreover, recent work has found evidence that ensembles may not be fundamental for uncertainty but simply better predictive models [Abe et al., 2022, Cattelan and Silva, 2022, Xia and Bouganis, 2022]. Thus, we do not consider ensembles here.

In this work we focus on simple post-hoc confidence estimators for softmax networks that can be directly computed from the logits. The earliest example of such a post-hoc method used for selective classification in a real-world application seems to be the use of LogitsMargin in [Le Cun et al., 1990]. While potentially suboptimal, such methods are extremely simple to apply on top of any trained classifier and should be natural choice to try before any more complex technique. In fact, it is not entirely obvious how a training-based approach should be compared to a post-hoc method. For instance, Feng et al. [2023] has found that, for some state-of-the-art training-based approaches to selective classification, *after* the main classifier has been trained with the corresponding technique, better selective classification performance can be obtained by discarding the auxiliary output providing confidence values and simply use the conventional MSP as the confidence estimator. Thus, in this sense, the MSP can be seen as a strong baseline.

Post-hoc methods have been widely considered in the context of calibration, among which the most popular approach is temperature scaling (TS). Applying TS to improve calibration (of the MSP confidence estimator) was originally proposed in [Guo et al., 2017] based on the negative log-likelihood. Optimizing TS for other metrics has been explored in [Mukhoti et al., 2020, Karandikar et al., 2021, Clarté et al., 2023] for calibration and in [Liang et al., 2023] for OOD detection, but had not been proposed for selective classification. A generalization of TS is adaptive TS (ATS) [Balanya et al., 2023], which uses an input-dependent temperature based on logits. The post-hoc methods we consider here can be seen as a special case of ATS, as logit norms may be seen as an input-dependent temperature; however Balanya et al. [2023] investigate a different temperature function and focuses on calibration. Other logit-based confidence estimators proposed for calibration and OOD detection include [Liu et al., 2020, Tomani et al., 2022, Rahimi et al., 2022, Neumann et al., 2018, Gonsior et al., 2022].

Normalizing the logits with the L_2 norm before applying the softmax function was used in [Kornblith et al., 2021] and later proposed and studied in [Wei et al., 2022] as a training technique (combined with TS) to improve OOD detection and calibration. A variation where the logits are normalized to unit variance was proposed in [Jiang et al., 2023] to accelerate training.

Benchmarking of models in their performance at selective classification/misclassification detection has been done in [Galil et al., 2023, Ding et al., 2020], however these works mostly consider the MSP as the confidence estimator. In the context of calibration, Wang et al. [2021] and Ashukha et al. [2020] have argued that models should be compared after simple post-hoc optimizations, since models that appear worse than others can sometimes easily be improved by methods such as TS. Here we advocate and provide further evidence for this approach in the context of selective classification.

B Confidence estimation

Besides the MSP, widely used as a baseline and presented in Section 2.2, other functions of the logits can be considered. Some examples are the *softmax margin* [Belghazi and Lopez-Paz, 2021, Lubrano et al., 2023], the *max logit* [Hendrycks et al., 2022], the *logits margin* [Streeter, 2018, Lebovitz et al., 2023], the *negative entropy*[1] [Belghazi and Lopez-Paz, 2021], and the *negative Gini index* [Granese et al., 2021, Gomes et al., 2022], defined, respectively, as

$$\text{SoftmaxMargin}(\mathbf{z}) \triangleq \sigma_{\hat{y}}(\mathbf{z}) - \max_{k\in\mathcal{Y}:k\neq\hat{y}} \sigma_k(\mathbf{z}) \tag{4}$$

$$\text{MaxLogit}(\mathbf{z}) \triangleq z_{\hat{y}} \tag{5}$$

$$\text{LogitsMargin}(\mathbf{z}) \triangleq z_{\hat{y}} - \max_{k\in\mathcal{Y}:k\neq\hat{y}} z_k \tag{6}$$

$$\text{NegativeEntropy}(\mathbf{z}) \triangleq \sum_{k\in\mathcal{Y}} \sigma_k(\mathbf{z}) \log \sigma_k(\mathbf{z}) \tag{7}$$

$$\text{NegativeGini}(\mathbf{z}) \triangleq -1 + \sum_{k\in\mathcal{Y}} \sigma_k(\mathbf{z})^2. \tag{8}$$

C MSP Fallback

A useful property of MSP-TS-AURC (but not MSP-TS-NLL) is that it can never have a worse performance than the MSP baseline, as long as $T = 1$ is included in the search space. It is natural to extend this property to every confidence estimator, for a simple reason: it is very easy to check whether the estimator provides an improvement to the MSP baseline and, if not, then use the MSP instead. Formally, this corresponds to adding a binary hyperparameter indicating an MSP fallback.

Equivalently, when measuring performance across different models, we simply report a (non-negligible) positive gain in NAURC whenever it occurs. More precisely, we define the *average positive gain* (APG) in NAURC as

$$\text{APG}(g) = \frac{1}{|\mathcal{H}|} \sum_{h\in\mathcal{H}} [\text{NAURC}(h, \text{MSP}) - \text{NAURC}(h, g)]_\epsilon^+, \qquad [x]_\epsilon^+ = \begin{cases} x, & \text{if } x > \epsilon \\ 0, & \text{otherwise} \end{cases}$$

where $\mathcal{H}$ is a set of classifiers and $\epsilon > 0$ is chosen so that only non-negligible gains are reported.

D Experiments

All the experiments in this section were performed using PyTorch [Paszke et al., 2019] and all of its provided classifiers pre-trained on ImageNet [Deng et al., 2009]. Additionally, some models of the Wightman [2019] repository were utilized, particularly the ones highlighted by Galil et al. [2023].

[1] Note that any uncertainty estimator can be used as a confidence estimator by taking its negative.

The list of the models, together with all the results per model are presented in Appendix I. In total, 84 ImageNet models were used for experiments. The validation set of ImageNet was randomly split into 5000 hold-out images for post-hoc optimization and 45000 for tests and comparisons. Investigations on the stability of this split are presented in Section G.

To give evidence that our results are not specific to ImageNet, we also run experiments on CIFAR-100 [Krizhevsky, 2009] and Oxford-IIIT Pet [Parkhi et al., 2012] datasets, which are presented in Appendix F.

D.1 Comparison of methods

We start by evaluating the NAURC of each possible combination of a confidence estimator listed in Section 2.2 with a logit transformation described in Section 4.2, for specific models. Table 2 and Table 3 shows the results for EfficientNet-V2-XL (trained on ImageNet-21K and fine tuned on ImageNet-1K) and VGG16, respectively, the former chosen for having the worst confidence estimator (in terms of AUROC) reported in Galil et al. [2023] and the latter chosen as a representative example of a lower accuracy model with a good confidence estimator.

Table 2: NAURC for post-hoc methods applied to an EfficientNet-V2-XL on ImageNet

	Logit Transformation			
Confidence Estimator	Raw	TS-NLL	TS-AURC	pNorm
MSP	0.4401	0.3504	0.2056	0.1714
SoftmaxMargin	0.3816	0.3143	0.2033	0.1705
MaxLogit	0.7695	–	–	**0.1684**
LogitsMargin	0.1935	–	–	0.1720
NegativeEntropy	0.5964	0.4286	0.2012	0.1715
NegativeGini	0.4485	0.3514	0.2067	0.1712

Table 3: NAURC for post-hoc methods applied to VGG16 on ImageNet

	Logit Transformation			
Confidence Estimator	Raw	TS-NLL	TS-AURC	pNorm
MSP	0.1838	0.1850	0.1836	0.1836
SoftmaxMargin	0.1898	0.1889	0.1887	0.1887
MaxLogit	0.3375	–	–	0.2012
LogitsMargin	0.2047	–	–	0.2047
NegativeEntropy	0.1968	0.2055	0.1837	0.1837
NegativeGini	0.1856	0.1888	0.1837	0.1837

As can be seen, on EfficientNet-V2-XL, the baseline MSP is easily outperformed by most methods. Surprisingly, the best method is not to use a softmax function but, instead, take the maximum of a p-normalized logit vector, leading to a reduction in NAURC of 0.27 points or about 62%.

However, on VGG16, the situation is quite different, as methods that use the unnormalized logits and improve the performance on EfficientNet-V2-XL, such as LogitsMargin and MaxLogit-pNorm, actually degrade it on VGG16. Moreover, the highest improvement obtained, e.g., with MSP-TS-AURC, is so small that it can be considered negligible. (In fact, gains below 0.003 NAURC are visually imperceptible in an AURC curve.) Thus, it is reasonable to assert that none of the post-hoc methods considered is able to outperform the baseline in this case.

Note that the baseline VGG16 and the optimized EfficientNet-V2-XL have NAURC values on a similar range, so we can interpret that VGG16 has innately already a good confidence estimator.

In Table 4, we evaluate the average performance of post-hoc methods across all models considered, using the APG-NAURC metric described in Section C, where we assume $\epsilon = 0.01$. Figure 3 shows the gains for selected methods for each model, ordered by MaxLogit-pNorm gains. It can be seen that the highest gains are provided by MaxLogit-pNorm, MSP-pNorm, NegativeGini-pNorm and their performance is essentially indistinguishable whenever they provide a non-negligible gain over the baseline. Moreover, the set of models for which significant gains can be obtained appears to be consistent across all methods.

Table 4: APG-NAURC of post-hoc methods across 84 ImageNet classifiers

	Logit Transformation			
Confidence Estimator	Raw	TS-NLL	TS-AURC	pNorm
MSP	0.0	0.03643	0.05776	0.06781
SoftmaxMargin	0.01966	0.04093	0.05597	0.06597
MaxLogit	0.0	–	–	**0.06837**
LogitsMargin	0.05501	–	–	0.06174
NegativeEntropy	0.0	0.01559	0.05904	0.06745
NegativeGini	0.0	0.03615	0.05816	0.06790

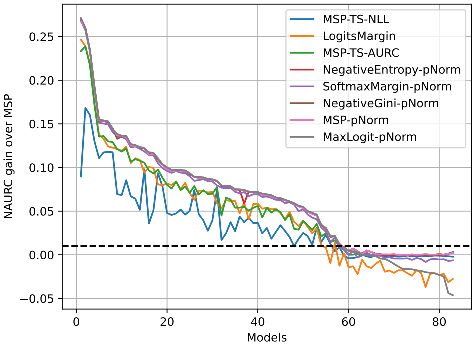


Figure 3: NAURC gains for post-hoc methods across 84 ImageNet classifiers. The dashed line denotes $\epsilon = 0.01$.

Although several post-hoc methods provide considerable gains, they all share a practical limitation which is the requirement of hold-out data for hyperparameter tuning. (A notable exception is LogitsMargin, which provides significant gains without having any hyperparameter.) In Appendix G, we study the data efficiency of some of the best performing methods. MaxLogit-pNorm, having a single hyperparameter, emerges as a clear winner, requiring fewer than 500 samples to achieve near-optimal performance on ImageNet (< 0.5 images per class on average) and fewer than 100 samples on CIFAR-100 (< 1 image per class on average). These requirements are clearly easily satisfied in practice for typical validation set sizes.

Details on the optimization of T and p, additional results showing AUROC values and RC curves, and results on the insensitivity of our conclusions to the choice of ϵ are provided in Appendix E.

D.2 Post-hoc optimization fixes broken confidence estimators

From Figure 3, we can distinguish two groups of models: those for which the MSP baseline is already the best confidence estimator and those for which post-hoc methods provide considerable gains (particularly, MaxLogit-pNorm). In fact, most models belong to the second group, comprising 58 out of 84 models considered.

Figure 4 illustrates two noteworthy phenomena. First, as previously observed by Galil et al. [2023], certain models exhibit superior accuracy than others but poorer uncertainty estimation, leading to a trade-off when selecting a classifier for selective classification. Second, post-hoc optimization can fix any "broken" confidence estimators.

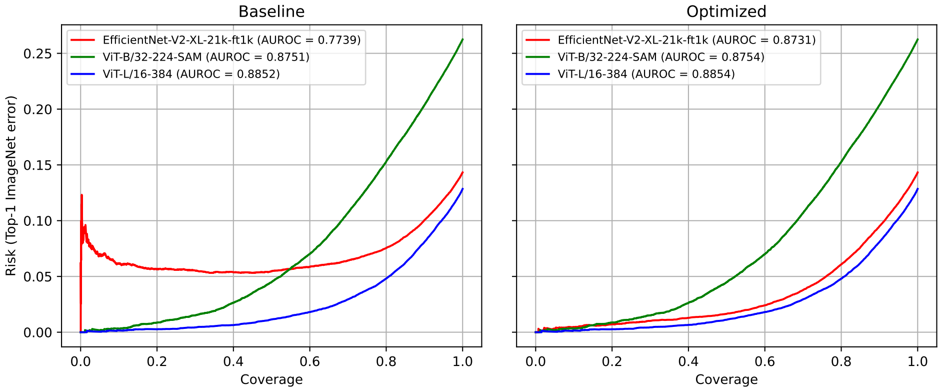


Figure 4: A comparison of RC curves made by three models selected in [Galil et al., 2023], including examples of highest (ViT-L/16-384) and lowest (EfficientNet-V2-XL) AUROC. After the application of a simple post-hoc method, the apparent pathology in EfficientNet-V2-XL completely disappears, resulting in significantly improved selective classification performance.

E More details and results on the experiments on ImageNet

E.1 Hyperparameter optimization of post-hoc methods

For not being differentiable, the NAURC metric demands a zero-order optimization. For this work, the optimizations of p and T were conducted via grid-search. Note that, as p approaches infinity, $||\mathbf{z}||_p \to \max(|\mathbf{z}|)$. Indeed, it tends to converge reasonable quickly. Thus, the grid search on p can be made only for small p. In our experiments, we noticed that it suffices to evaluate a few values of p, such as the integers between 0 and 10, where the 0-norm is taken here to mean the sum of all nonzero values of the vector. The temperature values were taken from the range between 0.01 and 3, with a step size of 0.01, as this showed to be sufficient for achieving the optimal temperature for selective classification (in general between 0 and 1).

E.2 AUROC results

Table 5 shows the AUROC results for all methods for an EfficientNetV2-XL on ImageNet, while Table 6 shows the same but for a VGG-16. As it can be seen, the results are consistent with the ones for NAURC presented in Section D.

Table 5: AUROC[x100] for post-hoc methods for an EfficientNet-V2-XL for ImageNet

	Logit Transformation			
Confidence Estimation	Raw	TS-NLL	TS-AUROC	pNorm
MSP	0.7732	0.8109	0.8587	0.8708
SoftmaxMargin	0.7990	0.8246	0.8590	0.8718
MaxLogit	0.6347	0.6347	0.6347	0.8741
LogitsMargin	0.8604	0.8604	0.8604	0.8701
NegativeEntropy	0.6890	0.7710	0.7538	0.8238
NegativeGini	0.7669	0.8101	0.8588	0.8711

E.3 RC curves

In Figure 5 the RC curves of selected post-hoc methods applied to a few representative models are shown.

Table 6: AUROC[x100] for post-hoc methods for VGG16 for ImageNet

	Logit Transformation			
Confidence Estimation	Raw	TS-NLL	TS-AUROC	pNorm
MSP	0.8661	0.8652	0.8662	0.8662
SoftmaxMargin	0.8603	0.8610	0.8617	0.8617
MaxLogit	0.7884	0.7884	0.7884	0.8557
LogitsMargin	0.8478	0.8478	0.8478	0.8478
NegativeEntropy	0.8556	0.8492	0.8659	0.8659
NegativeGini	0.8645	0.8619	0.8661	0.8661

E.4 Effect of ϵ

Figure 6 shows the results (in APG metric) for all methods when p is optimized. As can be seen, MaxLogit-pNorm is dominant for all $\epsilon > 0$, indicating that, provided the MSP fallback described in Section C is enabled, it outperforms the other methods.

F Experiments on additional datasets

F.1 Experiments on Oxford-IIIT Pet

The hold-out set for Oxford-IIIT Pet, consisting of 500 samples, was taken from the training set before training. The model used was an EfficientNet-V2-XL pretrained on ImageNet from Wightman [2019]. It was fine-tuned on Oxford-IIIT Pet [Parkhi et al., 2012]. The training was conducted for 100 epochs with Cross Entropy Loss, using a SGD optimizer with initial learning rate of 0.1 and a Cosine Annealing learning rate schedule with period 100. Moreover, a weight decay of 0.0005 and a Nesterov's momentum of 0.9 were used. Data transformations were applied, specifically standardization, random crop (for size 224x224) and random horizontal flip.

Figure 7 shows the RC curves for some selected methods for the EfficientNet-V2-XL. As can be seen, considerable gains are obtained with the optimization of p, especially in the low-risk region.

F.2 Experiments on CIFAR-100

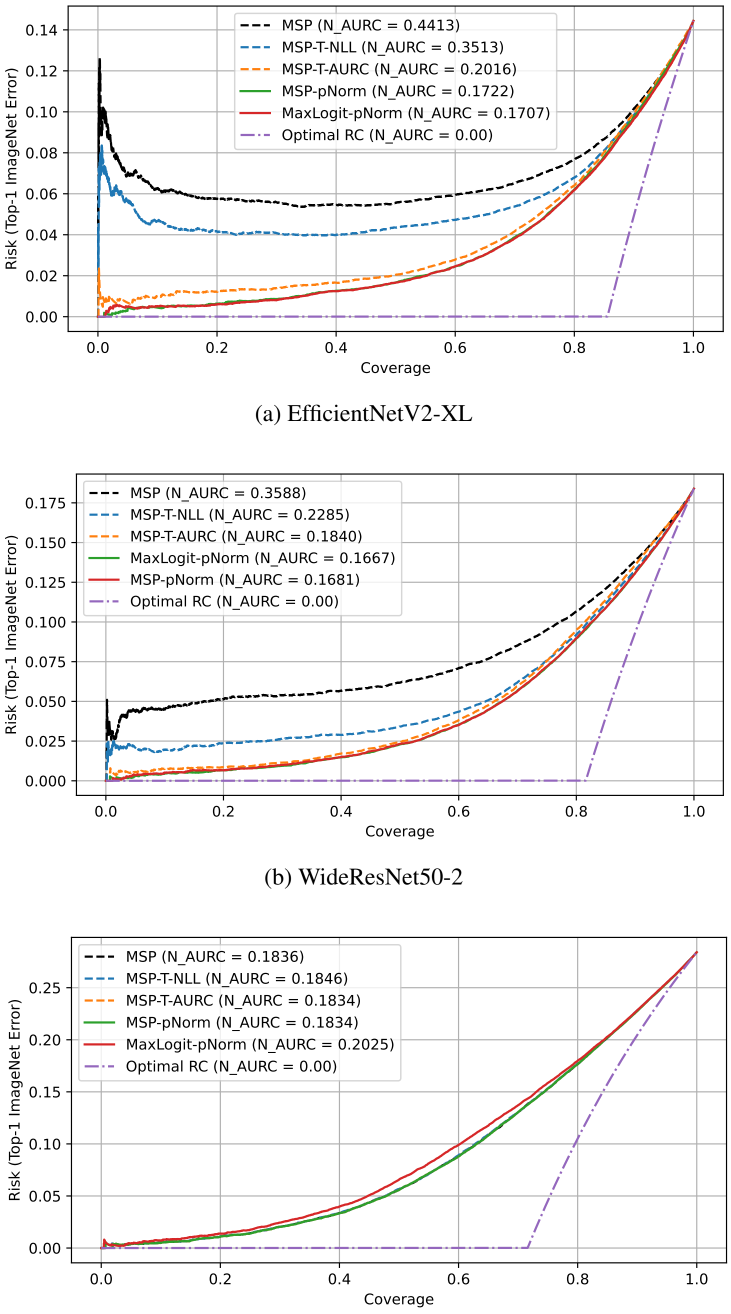
The hold-out set for CIFAR-100, consisting of 5000 samples, was taken from the training set before training. The model used was forked from `github.com/kuangliu/pytorch-cifar`, and adapted for CIFAR-100 [Krizhevsky, 2009]. It was trained for 200 epochs with Cross Entropy Loss, using a SGD optimizer with initial learning rate of 0.1 and a Cosine Annealing learning rate schedule with period 200. Moreover, a weight decay of 0.0005 and a Nesterov's momentum of 0.9 were used. Data transformations were applied, specifically standardization, random crop (for size 32x32 with padding 4) and random horizontal flip.

Figure 8 shows the RC curves for some selected methods for a VGG19. As it can be seen, the results follow the same pattern of the ones observed for ImageNet, with MaxLogit-pNorm achieving the best results.

G Data Efficiency

As mentioned in Section D, the experiments conducted in ImageNet used a hold-out dataset of 5,000 images randomly sampled from the validation dataset, resulting in 45,000 images reserved for the test phase.

In this section, the primary aim is to investigate the data efficiency of the methods, which indicates their capacity to learn and generalize from limited data. To accomplish this, the optimization process was executed multiple times, utilizing different fractions of the hold-out set while keeping the test set fixed at 45,000 samples. Consequently, two distinct types of random splits were implemented using the validation dataset. The first involved dividing the validation set into hold-out and test sets, while the second involved sampling fractions from the hold-out set. To ensure the findings



(a) EfficientNetV2-XL

(b) WideResNet50-2

(c) VGG16

Figure 5: RC curves for selected post-hoc methods applied to ImageNet classifiers.

were generalizable and robust, both of these random split procedures were repeated five times each, culminating in a total of 25 experiments for each analyzed fraction of the hold-out set.

Figure 9 displays the outcomes of these studies for an ResNet50 trained on ImageNet. As observed, MaxLogit-pNorm exhibits outstanding data efficiency, while methods that use temperature achieve lower efficiency.

Additionally, the data efficiency experiment was conducted on the VGG19 model for CIFAR-100. Indeed, the same conclusions hold about the high efficiency of MaxLogit-pNorm.

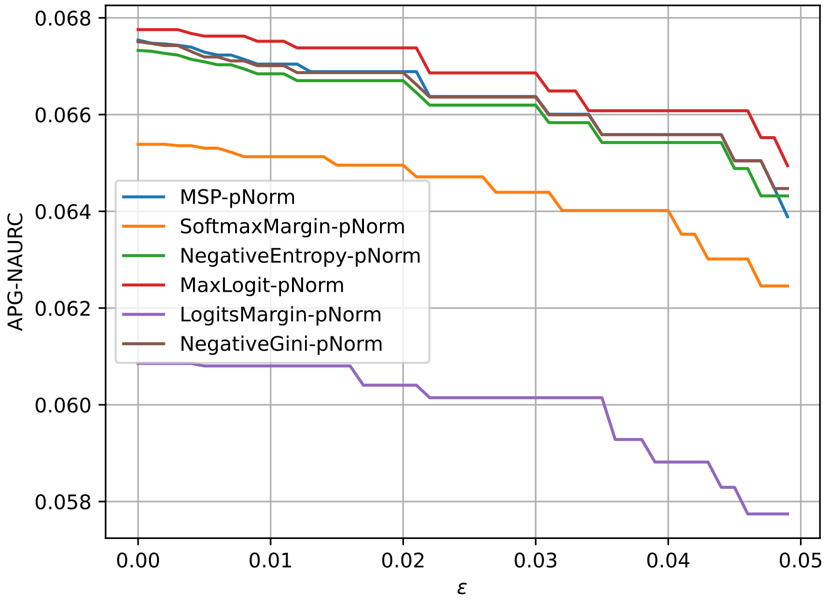


Figure 6: APG in terms of ϵ

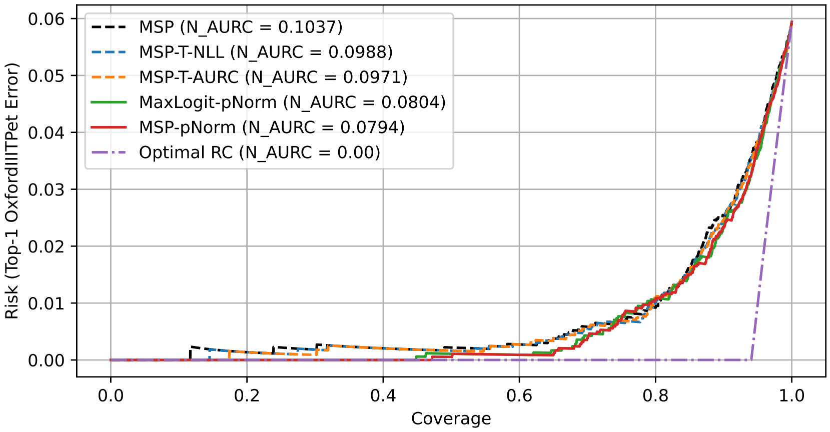


Figure 7: RC curves for a EfficientNet-V2-XL for Oxford-IIIT Pet

H More results on distribution shift

Figure 11 provides further insights into our conclusion regarding the strong correlation between NAURC and accuracy, even in the presence of distribution shifts. In this section, we extend our analysis beyond the findings outlined in Section 5 presenting the NAURC scores for ResNet50, WideResNet50-2, AlexNet, and ConvNext-Large models tested on ImageNet-C. These additional results consistently reinforce the conclusions drawn earlier.

I Full Results on ImageNet

Table 7 presents all the NAURC results for the most relevant methods for all the evaluated models on ImageNet, while Table 8 shows the corresponding AURC results and Table 9 the corresponding AUROC results.

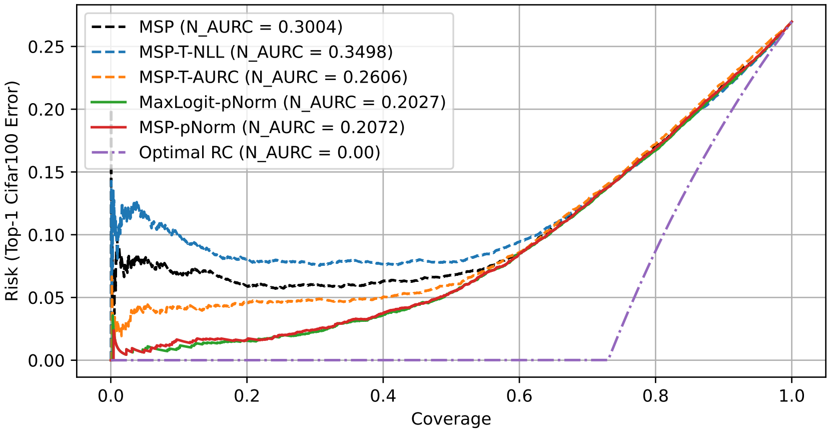


Figure 8: RC curves for a VGG19 for CIFAR-100

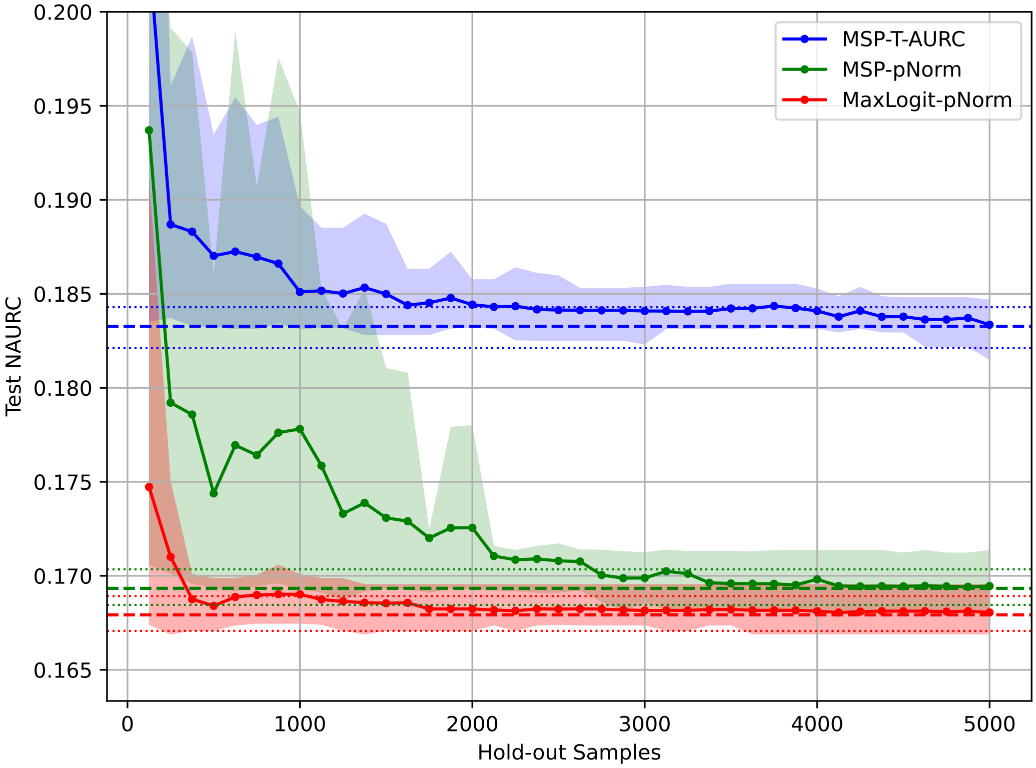


Figure 9: Data Efficiency: Average NAURC variation with number of hold-out samples used, for a ResNet50 on ImageNet. Dashed lines represent the optimal NAURC for each method, i.e., the achieved value when the optimization is made directly on the test set. Filled regions for each curve correspond to percentiles 10 and 90, while dotted lines represent the same but for the optimal value (optimized on test set directly).

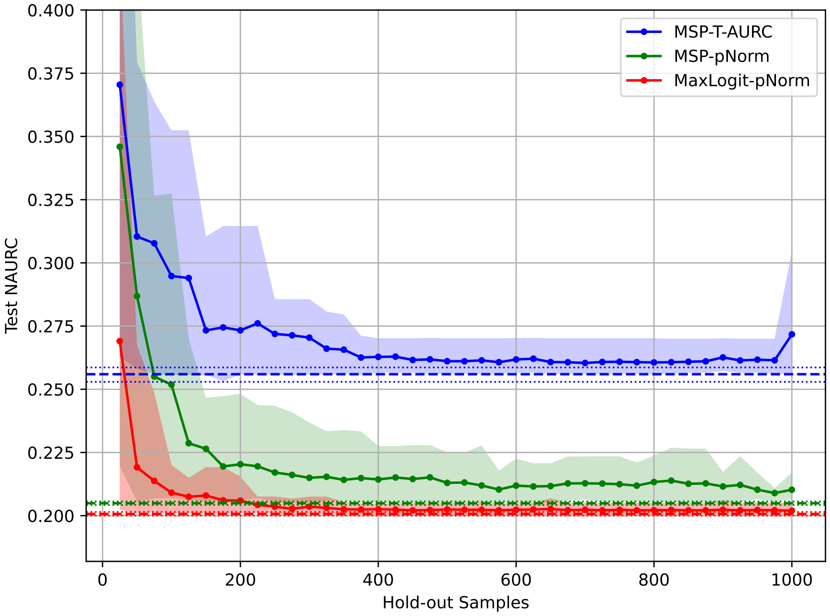
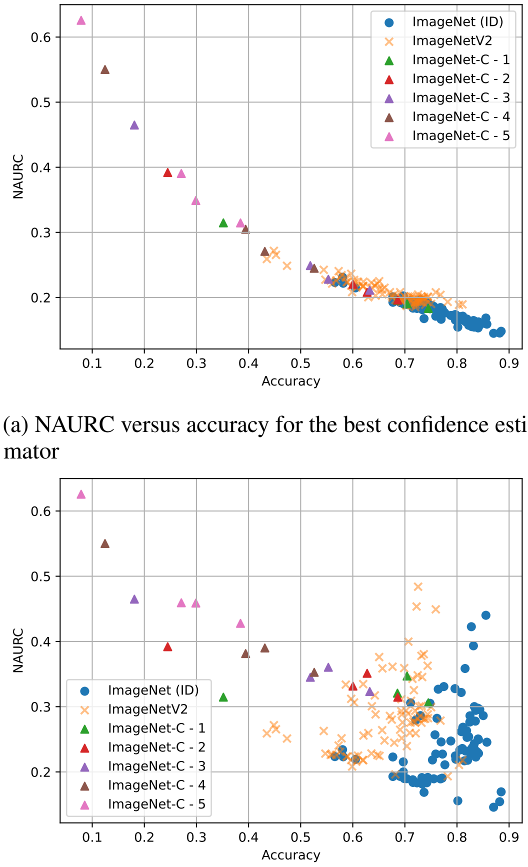


Figure 10: Data Efficiency: Average NAURC variation with number of hold-out samples used, for a VGG19 for CIFAR100. Dashed lines represent the optimal NAURC for each method, i.e., the achieved value when the optimization is made directly on the test set. Filled regions for each curve correspond to percentiles 10 and 90, while dotted lines represent the same but for the optimal value (optimized on test set directly).



(a) NAURC versus accuracy for the best confidence estimator

(b) NAURC versus accuracy for the MSP baseline

Figure 11: NAURC under distribution shift for all models considered for ImageNet

Table 7: NAURC for all models evaluated on ImageNet

		Method					
Model	Accuracy[%]	MSP	MSP-TS-NLL	MSP-TS-AURC	LogitsMargin	MSP-pNorm - p^*	MaxLogit-pNorm - p^*
alexnet	0.5657	0.2234	0.2248	0.2232	0.2604	0.2232 - 0	0.2401 - 0
convnext_base	0.8402	0.2977	0.2293	0.1795	0.1784	0.1613 - 4	0.1611 - 5
convnext_large	0.8438	0.2956	0.2413	0.1772	0.1728	0.1582 - 5	0.1574 - 5
convnext_small	0.8360	0.2944	0.2249	0.1733	0.1732	0.1580 - 3	0.1558 - 5
convnext_tiny	0.8249	0.2860	0.2190	0.1832	0.1810	0.1605 - 5	0.1592 - 6
densenet121	0.7439	0.1915	0.1926	0.1915	0.2095	0.1914 - 0	0.1989 - 0
densenet161	0.7715	0.1904	0.1943	0.1903	0.2036	0.1831 - 3	0.1870 - 7
densenet169	0.7555	0.1895	0.1923	0.1894	0.2046	0.1862 - 1	0.1907 - 7
densenet201	0.7683	0.1894	0.1911	0.1887	0.2024	0.1844 - 3	0.1894 - 7
efficientnet_b0	0.7768	0.2110	0.1962	0.1965	0.2004	0.1761 - 4	0.1764 - 6
efficientnet_b1	0.7977	0.2194	0.1884	0.1819	0.1891	0.1745 - 4	0.1733 - 6
efficientnet_b2	0.8054	0.2336	0.2062	0.1868	0.1929	0.1714 - 5	0.1705 - 6
efficientnet_b3	0.8193	0.2529	0.2066	0.1810	0.1794	0.1651 - 5	0.1637 - 6
efficientnet_b4	0.8333	0.3001	0.2147	0.1750	0.1763	0.1692 - 3	0.1660 - 7
efficientnet_b5	0.8334	0.2350	0.1977	0.1704	0.1736	0.1574 - 4	0.1563 - 6
efficientnet_b6	0.8389	0.2304	0.1930	0.1735	0.1705	0.1735 - 0	0.1559 - 0
efficientnet_b7	0.8406	0.2505	0.2031	0.1639	0.1664	0.1550 - 3	0.1532 - 6
efficientnet_v2_l	0.8574	0.2456	0.2061	0.1730	0.1734	0.1597 - 5	0.1590 - 6
efficientnet_v2_m	0.8504	0.2863	0.2220	0.1768	0.1757	0.1646 - 3	0.1606 - 5
efficientnet_v2_s	0.8413	0.2284	0.1918	0.1667	0.1698	0.1563 - 4	0.1556 - 5
googlenet	0.6970	0.2154	0.2068	0.2052	0.2279	0.2024 - 3	0.2040 - 6
inception_v3	0.7722	0.2273	0.2158	0.1973	0.2026	0.1800 - 4	0.1791 - 5
maxvit_t	0.8362	0.2228	0.2024	0.1737	0.1735	0.1614 - 5	0.1600 - 5
mnasnet0_5	0.6769	0.2229	0.2104	0.2086	0.2324	0.2014 - 3	0.2013 - 7
mnasnet0_75	0.7110	0.3059	0.2124	0.2081	0.2252	0.1955 - 3	0.1964 - 6
mnasnet1_0	0.7349	0.1833	0.1854	0.1834	0.2010	0.1833 - 0	0.1917 - 0
mnasnet1_3	0.7640	0.3268	0.2093	0.1960	0.2039	0.1814 - 4	0.1815 - 6
mobilenet_v2	0.7211	0.2795	0.2049	0.2020	0.2204	0.1943 - 4	0.1952 - 6
mobilenet_v3_large	0.7529	0.2183	0.1953	0.1926	0.2099	0.1874 - 5	0.1873 - 6
mobilenet_v3_small	0.6770	0.1929	0.1946	0.1926	0.2170	0.1926 - 0	0.2047 - 0

regnet_x_16gf	0.8266	0.2271	0.2008	0.1743	0.1745	0.1619 - 4	0.1613 - 5
regnet_x_1_6gf	0.7962	0.3277	0.2104	0.1899	0.1948	0.1740 - 4	0.1736 - 6
regnet_x_32gf	0.8294	0.2373	0.2127	0.1719	0.1746	0.1621 - 3	0.1583 - 5
regnet_x_3_2gf	0.8111	0.2679	0.2161	0.1905	0.1928	0.1713 - 5	0.1718 - 6
regnet_x_400mf	0.7483	0.3052	0.2072	0.1989	0.2111	0.1826 - 4	0.1836 - 6
regnet_x_800mf	0.7748	0.3308	0.2128	0.1986	0.2067	0.1778 - 4	0.1779 - 5
regnet_x_8gf	0.8166	0.2320	0.2013	0.1754	0.1778	0.1653 - 3	0.1626 - 5
regnet_y_128gf	0.8819	0.1543	0.1553	0.1489	0.1525	0.1463 - 4	0.1455 - 7
regnet_y_16gf	0.8284	0.2786	0.2271	0.1726	0.1700	0.1553 - 4	0.1551 - 5
regnet_y_1_6gf	0.8085	0.2617	0.2113	0.1893	0.1910	0.1684 - 4	0.1682 - 5
regnet_y_32gf	0.8332	0.2432	0.2038	0.1693	0.1701	0.1555 - 3	0.1544 - 5
regnet_y_3_2gf	0.8189	0.2318	0.1980	0.1811	0.1841	0.1669 - 6	0.1670 - 5
regnet_y_400mf	0.7578	0.2569	0.2144	0.2047	0.2164	0.1839 - 4	0.1847 - 5
regnet_y_800mf	0.7881	0.2474	0.2036	0.1909	0.1999	0.1725 - 4	0.1724 - 5
regnet_y_8gf	0.8273	0.2291	0.1928	0.1703	0.1711	0.1576 - 5	0.1570 - 5
resnet101	0.8185	0.2632	0.2176	0.1825	0.1832	0.1675 - 5	0.1658 - 5
resnet152	0.8225	0.2540	0.2081	0.1702	0.1716	0.1584 - 4	0.1573 - 5
resnet18	0.6971	0.2000	0.2018	0.1994	0.2208	0.1999 - 1	0.2090 - 7
resnet34	0.7326	0.1913	0.1928	0.1914	0.2107	0.1907 - 2	0.1956 - 7
resnet50	0.8082	0.3218	0.2109	0.1814	0.1841	0.1679 - 4	0.1667 - 5
resnext101_32x8d	0.8276	0.4226	0.2542	0.1863	0.1836	0.1633 - 4	0.1633 - 6
resnext101_64x4d	0.8316	0.3933	0.2332	0.1763	0.1746	0.1609 - 3	0.1579 - 6
resnext50_32x4d	0.8116	0.2685	0.2206	0.1843	0.1870	0.1703 - 5	0.1683 - 5
shufflenet_v2_x0_5	0.6052	0.2192	0.2218	0.2180	0.2412	0.2147 - 4	0.2159 - 7
shufflenet_v2_x1_0	0.6924	0.1965	0.2005	0.1961	0.2107	0.1919 - 4	0.1918 - 7
shufflenet_v2_x1_5	0.7299	0.2863	0.2122	0.2072	0.2233	0.1963 - 4	0.1970 - 6
shufflenet_v2_x2_0	0.7616	0.2822	0.2041	0.1938	0.2024	0.1787 - 4	0.1782 - 6
squeezenet1_0	0.5804	0.2341	0.2363	0.2318	0.2618	0.2317 - 0	0.2751 - 0
squeezenet1_1	0.5817	0.2232	0.2249	0.2220	0.2546	0.2220 - 0	0.2632 - 0
swin_b	0.8353	0.2778	0.2420	0.1764	0.1773	0.1650 - 3	0.1604 - 5
swin_s	0.8316	0.2329	0.2145	0.1813	0.1808	0.1660 - 4	0.1645 - 5
swin_t	0.8143	0.2170	0.1956	0.1801	0.1850	0.1678 - 4	0.1668 - 5
swin_v2_b	0.8412	0.2494	0.2219	0.1757	0.1780	0.1633 - 4	0.1625 - 5
swin_v2_s	0.8365	0.2317	0.2047	0.1688	0.1702	0.1577 - 4	0.1565 - 5
swin_v2_t	0.8204	0.2178	0.1929	0.1747	0.1786	0.1637 - 4	0.1627 - 5
vgg11	0.6909	0.1931	0.1941	0.1926	0.2164	0.1926 - 0	0.2147 - 0

vgg11_bn	0.7034	0.1896	0.1909	0.1895	0.2114	0.1895 - 0	0.2134 - 0
vgg13	0.6988	0.1898	0.1907	0.1895	0.2108	0.1895 - 0	0.2100 - 0
vgg13_bn	0.7151	0.1880	0.1895	0.1878	0.2098	0.1877 - 0	0.2073 - 0
vgg16	0.7156	0.1838	0.1850	0.1836	0.2047	0.1836 - 0	0.2012 - 0
vgg16_bn	0.7335	0.1814	0.1833	0.1813	0.1993	0.1813 - 0	0.1962 - 0
vgg19	0.7232	0.1832	0.1841	0.1831	0.2044	0.1831 - 0	0.1988 - 0
vgg19_bn	0.7420	0.1834	0.1851	0.1834	0.2008	0.1834 - 0	0.2002 - 0
vit_b_16	0.8102	0.2359	0.2113	0.1815	0.1831	0.1669 - 4	0.1657 - 5
vit_b_32	0.7592	0.2274	0.2092	0.1903	0.1944	0.1719 - 4	0.1712 - 5
vit_h_14	0.8848	0.1695	0.1652	0.1533	0.1557	0.1483 - 4	0.1480 - 6
vit_l_16	0.7969	0.2246	0.2146	0.1841	0.1869	0.1652 - 4	0.1654 - 4
vit_l_32	0.7696	0.2460	0.2290	0.1932	0.1930	0.1664 - 4	0.1677 - 5
wide_resnet101_2	0.8246	0.2765	0.2255	0.1799	0.1769	0.1606 - 5	0.1595 - 5
wide_resnet50_2	0.8155	0.3585	0.2284	0.1830	0.1852	0.1663 - 4	0.1654 - 6
efficientnetv2_xl	0.8555	0.4401	0.3504	0.2056	0.1935	0.1714 - 4	0.1684 - 6
vit_l_16_384	0.8702	0.1459	0.1461	0.1451	0.1526	0.1450 - 0	0.1502 - 0
vit_b_16_sam	0.8016	0.1557	0.1555	0.1546	0.1614	0.1545 - 0	0.1560 - 0
vit_b_32_sam	0.7368	0.1687	0.1682	0.1676	0.1789	0.1676 - 0	0.1693 - 0

Table 8: AURC for all models evaluated on ImageNet

		Method					
Model	Accuracy[%]	MSP	MSP-TS-NLL	MSP-TS-AURC	LogitsMargin	MSP-pNorm - p^*	MaxLogit-pNorm - p^*
alexnet	0.5657	0.1840	0.1845	0.1840	0.1959	0.1839 - 0	0.1894 - 0
convnext_base	0.8402	0.0570	0.0470	0.0398	0.0396	0.0371 - 4	0.0370 - 5
convnext_large	0.8438	0.0553	0.0475	0.0383	0.0377	0.0355 - 5	0.0354 - 5
convnext_small	0.8360	0.0583	0.0479	0.0402	0.0402	0.0379 - 3	0.0376 - 5
convnext_tiny	0.8249	0.0617	0.0511	0.0454	0.0450	0.0417 - 5	0.0415 - 6
densenet121	0.7439	0.0782	0.0784	0.0781	0.0821	0.0781 - 0	0.0797 - 0
densenet161	0.7715	0.0665	0.0672	0.0665	0.0691	0.0650 - 3	0.0658 - 7
densenet169	0.7555	0.0728	0.0734	0.0728	0.0760	0.0721 - 1	0.0730 - 7
densenet201	0.7683	0.0675	0.0679	0.0674	0.0702	0.0665 - 3	0.0675 - 7
efficientnet_b0	0.7768	0.0684	0.0655	0.0656	0.0663	0.0615 - 4	0.0616 - 6
efficientnet_b1	0.7977	0.0616	0.0560	0.0548	0.0561	0.0534 - 4	0.0532 - 6
efficientnet_b2	0.8054	0.0610	0.0562	0.0529	0.0539	0.0501 - 5	0.0500 - 6
efficientnet_b3	0.8193	0.0587	0.0512	0.0470	0.0467	0.0443 - 5	0.0441 - 6
efficientnet_b4	0.8333	0.0604	0.0474	0.0413	0.0415	0.0404 - 3	0.0399 - 7
efficientnet_b5	0.8334	0.0504	0.0447	0.0406	0.0411	0.0386 - 4	0.0384 - 6
efficientnet_b6	0.8389	0.0477	0.0422	0.0393	0.0389	0.0393 - 0	0.0367 - 0
efficientnet_b7	0.8406	0.0500	0.0431	0.0374	0.0377	0.0360 - 3	0.0358 - 6
efficientnet_v2_l	0.8574	0.0431	0.0379	0.0335	0.0336	0.0317 - 5	0.0316 - 6
efficientnet_v2_m	0.8504	0.0513	0.0424	0.0362	0.0360	0.0345 - 3	0.0339 - 5
efficientnet_v2_s	0.8413	0.0465	0.0412	0.0376	0.0380	0.0360 - 4	0.0359 - 5
googlenet	0.6970	0.1056	0.1034	0.1030	0.1088	0.1023 - 3	0.1027 - 6
inception_v3	0.7722	0.0736	0.0713	0.0676	0.0686	0.0641 - 4	0.0639 - 5
maxvit_t	0.8362	0.0476	0.0445	0.0402	0.0402	0.0383 - 5	0.0381 - 5
mnasnet0_5	0.6769	0.1178	0.1146	0.1141	0.1204	0.1121 - 3	0.1121 - 7
mnasnet0_75	0.7110	0.1207	0.0980	0.0970	0.1011	0.0939 - 3	0.0941 - 6
mnasnet1_0	0.7349	0.0802	0.0807	0.0803	0.0843	0.0802 - 0	0.0821 - 0
mnasnet1_3	0.7640	0.0975	0.0734	0.0706	0.0723	0.0676 - 4	0.0676 - 6
mobilenet_v2	0.7211	0.1090	0.0914	0.0908	0.0951	0.0889 - 4	0.0891 - 6
mobilenet_v3_large	0.7529	0.0801	0.0752	0.0746	0.0783	0.0734 - 5	0.0734 - 6
mobilenet_v3_small	0.6770	0.1099	0.1103	0.1098	0.1162	0.1097 - 0	0.1129 - 0

regnet_x_16gf	0.8266	0.0517	0.0476	0.0434	0.0435	0.0414 - 4	0.0413 - 5
regnet_x_1_6gf	0.7962	0.0818	0.0605	0.0568	0.0577	0.0539 - 4	0.0538 - 6
regnet_x_32gf	0.8294	0.0523	0.0485	0.0421	0.0425	0.0406 - 3	0.0400 - 5
regnet_x_3_2gf	0.8111	0.0646	0.0558	0.0514	0.0518	0.0481 - 5	0.0482 - 6
regnet_x_400mf	0.7483	0.1009	0.0797	0.0779	0.0805	0.0743 - 4	0.0745 - 6
regnet_x_800mf	0.7748	0.0929	0.0696	0.0668	0.0684	0.0626 - 4	0.0626 - 5
regnet_x_8gf	0.8166	0.0563	0.0513	0.0470	0.0474	0.0452 - 3	0.0448 - 5
regnet_y_128gf	0.8819	0.0244	0.0245	0.0238	0.0242	0.0234 - 4	0.0233 - 7
regnet_y_16gf	0.8284	0.0591	0.0511	0.0426	0.0422	0.0398 - 4	0.0398 - 5
regnet_y_1_6gf	0.8085	0.0646	0.0559	0.0522	0.0524	0.0485 - 4	0.0485 - 5
regnet_y_32gf	0.8332	0.0517	0.0458	0.0405	0.0406	0.0384 - 3	0.0382 - 5
regnet_y_3_2gf	0.8189	0.0554	0.0499	0.0471	0.0476	0.0447 - 6	0.0448 - 5
regnet_y_400mf	0.7578	0.0861	0.0771	0.0751	0.0775	0.0707 - 4	0.0708 - 5
regnet_y_800mf	0.7881	0.0707	0.0624	0.0601	0.0617	0.0566 - 4	0.0565 - 5
regnet_y_8gf	0.8273	0.0518	0.0461	0.0426	0.0427	0.0405 - 5	0.0404 - 5
resnet101	0.8185	0.0607	0.0533	0.0475	0.0476	0.0450 - 5	0.0447 - 5
resnet152	0.8225	0.0576	0.0502	0.0441	0.0444	0.0422 - 4	0.0420 - 5
resnet18	0.6971	0.1017	0.1021	0.1015	0.1069	0.1016 - 1	0.1039 - 7
resnet34	0.7326	0.0830	0.0834	0.0831	0.0875	0.0829 - 2	0.0840 - 7
resnet50	0.8082	0.0751	0.0560	0.0509	0.0514	0.0486 - 4	0.0484 - 5
resnext101_32x8d	0.8276	0.0820	0.0556	0.0450	0.0446	0.0413 - 4	0.0413 - 6
resnext101_64x4d	0.8316	0.0754	0.0508	0.0421	0.0418	0.0397 - 3	0.0392 - 6
resnext50_32x4d	0.8116	0.0645	0.0563	0.0502	0.0507	0.0478 - 5	0.0474 - 5
shufflenet_v2_x0_5	0.6052	0.1575	0.1583	0.1571	0.1642	0.1561 - 4	0.1565 - 7
shufflenet_v2_x1_0	0.6924	0.1031	0.1041	0.1030	0.1067	0.1019 - 4	0.1019 - 7
shufflenet_v2_x1_5	0.7299	0.1061	0.0890	0.0879	0.0916	0.0854 - 4	0.0855 - 6
shufflenet_v2_x2_0	0.7616	0.0895	0.0733	0.0712	0.0730	0.0680 - 4	0.0679 - 6
squeezenet1_0	0.5804	0.1777	0.1784	0.1770	0.1865	0.1769 - 0	0.1906 - 0
squeezenet1_1	0.5817	0.1735	0.1740	0.1731	0.1834	0.1731 - 0	0.1861 - 0
swin_b	0.8353	0.0561	0.0508	0.0409	0.0410	0.0391 - 3	0.0384 - 5
swin_s	0.8316	0.0507	0.0479	0.0428	0.0428	0.0405 - 4	0.0402 - 5
swin_t	0.8143	0.0547	0.0511	0.0486	0.0494	0.0465 - 4	0.0463 - 5
swin_v2_b	0.8412	0.0496	0.0456	0.0389	0.0392	0.0371 - 4	0.0369 - 5
swin_v2_s	0.8365	0.0488	0.0447	0.0394	0.0396	0.0377 - 4	0.0375 - 5
swin_v2_t	0.8204	0.0526	0.0485	0.0456	0.0462	0.0437 - 4	0.0436 - 5
vgg11	0.6909	0.1030	0.1032	0.1029	0.1089	0.1028 - 0	0.1085 - 0

vgg11_bn	0.7034	0.0961	0.0964	0.0960	0.1015	0.0960 - 0	0.1019 - 0
vgg13	0.6988	0.0983	0.0985	0.0982	0.1036	0.0982 - 0	0.1033 - 0
vgg13_bn	0.7151	0.0902	0.0905	0.0901	0.0954	0.0901 - 0	0.0948 - 0
vgg16	0.7156	0.0890	0.0892	0.0889	0.0940	0.0889 - 0	0.0931 - 0
vgg16_bn	0.7335	0.0804	0.0808	0.0804	0.0845	0.0803 - 0	0.0837 - 0
vgg19	0.7232	0.0853	0.0856	0.0853	0.0903	0.0853 - 0	0.0890 - 0
vgg19_bn	0.7420	0.0772	0.0775	0.0772	0.0810	0.0771 - 0	0.0808 - 0
vit_b_16	0.8102	0.0595	0.0553	0.0502	0.0505	0.0477 - 4	0.0475 - 5
vit_b_32	0.7592	0.0792	0.0754	0.0714	0.0723	0.0675 - 4	0.0674 - 5
vit_h_14	0.8848	0.0253	0.0248	0.0235	0.0238	0.0229 - 4	0.0229 - 6
vit_l_16	0.7969	0.0628	0.0610	0.0555	0.0560	0.0520 - 4	0.0521 - 4
vit_l_32	0.7696	0.0784	0.0750	0.0678	0.0678	0.0624 - 4	0.0626 - 5
wide_resnet101_2	0.8246	0.0603	0.0522	0.0450	0.0445	0.0419 - 5	0.0417 - 5
wide_resnet50_2	0.8155	0.0778	0.0562	0.0486	0.0490	0.0458 - 4	0.0456 - 6
efficientnetv2_xl	0.8555	0.0698	0.0578	0.0384	0.0368	0.0338 - 4	0.0334 - 6
vit_l_16_384	0.8702	0.0265	0.0265	0.0264	0.0273	0.0263 - 0	0.0269 - 0
vit_b_16_sam	0.8016	0.0487	0.0487	0.0485	0.0497	0.0485 - 0	0.0487 - 0
vit_b_32_sam	0.7368	0.0761	0.0760	0.0759	0.0784	0.0758 - 0	0.0762 - 0

Table 9: AUROC for all models evaluated on ImageNet

		Method					
Model	Accuracy[%]	MSP	MSP-TS-NLL	MSP-TS-AURC	LogitsMargin	MSP-pNorm - p^*	MaxLogit-pNorm - p^*
alexnet	0.5657	0.8483	0.8475	0.8483	0.8181	0.8482 - 0	0.8391 - 0
convnext_base	0.8402	0.8255	0.8537	0.8672	0.8652	0.8769 - 4	0.8771 - 5
convnext_large	0.8438	0.8257	0.8496	0.8700	0.8686	0.8793 - 5	0.8796 - 5
convnext_small	0.8360	0.8271	0.8560	0.8702	0.8678	0.8780 - 3	0.8801 - 5
convnext_tiny	0.8249	0.8247	0.8560	0.8659	0.8633	0.8781 - 5	0.8786 - 6
densenet121	0.7439	0.8604	0.8596	0.8604	0.8454	0.8603 - 0	0.8564 - 0
densenet161	0.7715	0.8635	0.8612	0.8633	0.8520	0.8668 - 3	0.8643 - 7
densenet169	0.7555	0.8654	0.8634	0.8654	0.8520	0.8663 - 1	0.8631 - 7
densenet201	0.7683	0.8628	0.8616	0.8630	0.8512	0.8647 - 3	0.8619 - 7
efficientnet_b0	0.7768	0.8570	0.8642	0.8581	0.8535	0.8704 - 4	0.8701 - 6
efficientnet_b1	0.7977	0.8538	0.8672	0.8676	0.8584	0.8711 - 4	0.8713 - 6
efficientnet_b2	0.8054	0.8509	0.8628	0.8632	0.8592	0.8740 - 5	0.8740 - 6
efficientnet_b3	0.8193	0.8435	0.8636	0.8701	0.8660	0.8770 - 5	0.8775 - 6
efficientnet_b4	0.8333	0.8213	0.8600	0.8697	0.8668	0.8739 - 3	0.8750 - 7
efficientnet_b5	0.8334	0.8549	0.8691	0.8724	0.8693	0.8809 - 4	0.8817 - 6
efficientnet_b6	0.8389	0.8575	0.8714	0.8750	0.8704	0.8750 - 0	0.8811 - 0
efficientnet_b7	0.8406	0.8509	0.8676	0.8761	0.8730	0.8814 - 3	0.8827 - 6
efficientnet_v2_l	0.8574	0.8467	0.8645	0.8724	0.8700	0.8796 - 5	0.8798 - 6
efficientnet_v2_m	0.8504	0.8251	0.8549	0.8696	0.8680	0.8750 - 3	0.8779 - 5
efficientnet_v2_s	0.8413	0.8591	0.8718	0.8740	0.8707	0.8810 - 4	0.8815 - 5
googlenet	0.6970	0.8490	0.8542	0.8550	0.8361	0.8563 - 3	0.8552 - 6
inception_v3	0.7722	0.8491	0.8548	0.8605	0.8537	0.8700 - 4	0.8702 - 5
maxvit_t	0.8362	0.8594	0.8658	0.8698	0.8672	0.8776 - 5	0.8781 - 5
mnasnet0_5	0.6769	0.8465	0.8534	0.8536	0.8333	0.8577 - 3	0.8574 - 7
mnasnet0_75	0.7110	0.8025	0.8510	0.8526	0.8377	0.8592 - 3	0.8584 - 6
mnasnet1_0	0.7349	0.8659	0.8643	0.8658	0.8506	0.8658 - 0	0.8601 - 0
mnasnet1_3	0.7640	0.7942	0.8554	0.8592	0.8501	0.8665 - 4	0.8662 - 6
mobilenet_v2	0.7211	0.8167	0.8554	0.8559	0.8399	0.8600 - 4	0.8593 - 6
mobilenet_v3_large	0.7529	0.8519	0.8625	0.8620	0.8464	0.8638 - 5	0.8636 - 6
mobilenet_v3_small	0.6770	0.8621	0.8611	0.8622	0.8415	0.8622 - 0	0.8547 - 0

regnet_x_16gf	0.8266	0.8553	0.8664	0.8703	0.8675	0.8777 - 4	0.8781 - 5
regnet_x_1_6gf	0.7962	0.7991	0.8584	0.8627	0.8575	0.8718 - 4	0.8720 - 6
regnet_x_32gf	0.8294	0.8551	0.8643	0.8717	0.8685	0.8766 - 3	0.8794 - 5
regnet_x_3_2gf	0.8111	0.8340	0.8583	0.8634	0.8592	0.8722 - 5	0.8718 - 6
regnet_x_400mf	0.7483	0.8133	0.8583	0.8583	0.8457	0.8660 - 4	0.8654 - 6
regnet_x_800mf	0.7748	0.7981	0.8562	0.8593	0.8503	0.8691 - 4	0.8688 - 5
regnet_x_8gf	0.8166	0.8532	0.8656	0.8698	0.8660	0.8750 - 3	0.8774 - 5
regnet_y_128gf	0.8819	0.8836	0.8832	0.8835	0.8804	0.8857 - 4	0.8862 - 7
regnet_y_16gf	0.8284	0.8397	0.8590	0.8715	0.8697	0.8809 - 4	0.8810 - 5
regnet_y_1_6gf	0.8085	0.8396	0.8616	0.8660	0.8588	0.8742 - 4	0.8742 - 5
regnet_y_32gf	0.8332	0.8488	0.8642	0.8714	0.8684	0.8801 - 3	0.8810 - 5
regnet_y_3_2gf	0.8189	0.8522	0.8654	0.8672	0.8615	0.8739 - 6	0.8739 - 5
regnet_y_400mf	0.7578	0.8386	0.8575	0.8574	0.8448	0.8667 - 4	0.8661 - 5
regnet_y_800mf	0.7881	0.8413	0.8610	0.8613	0.8521	0.8715 - 4	0.8711 - 5
regnet_y_8gf	0.8273	0.8526	0.8683	0.8718	0.8687	0.8794 - 5	0.8797 - 5
resnet101	0.8185	0.8423	0.8604	0.8665	0.8631	0.8753 - 5	0.8758 - 5
resnet152	0.8225	0.8465	0.8652	0.8722	0.8691	0.8799 - 4	0.8808 - 5
resnet18	0.6971	0.8573	0.8561	0.8577	0.8398	0.8573 - 1	0.8521 - 7
resnet34	0.7326	0.8619	0.8608	0.8618	0.8456	0.8621 - 2	0.8592 - 7
resnet50	0.8082	0.8060	0.8602	0.8658	0.8617	0.8744 - 4	0.8745 - 5
resnext101_32x8d	0.8276	0.7680	0.8447	0.8654	0.8635	0.8768 - 4	0.8769 - 6
resnext101_64x4d	0.8316	0.7796	0.8534	0.8701	0.8684	0.8777 - 3	0.8799 - 6
resnext50_32x4d	0.8116	0.8360	0.8569	0.8644	0.8606	0.8735 - 5	0.8743 - 5
shufflenet_v2_x0_5	0.6052	0.8513	0.8498	0.8519	0.8319	0.8534 - 4	0.8524 - 7
shufflenet_v2_x1_0	0.6924	0.8602	0.8578	0.8602	0.8469	0.8629 - 4	0.8628 - 7
shufflenet_v2_x1_5	0.7299	0.8123	0.8518	0.8536	0.8390	0.8594 - 4	0.8587 - 6
shufflenet_v2_x2_0	0.7616	0.8170	0.8588	0.8617	0.8524	0.8693 - 4	0.8693 - 6
squeezenet1_0	0.5804	0.8424	0.8410	0.8437	0.8182	0.8436 - 0	0.8177 - 0
squeezenet1_1	0.5817	0.8486	0.8476	0.8492	0.8227	0.8492 - 0	0.8254 - 0
swin_b	0.8353	0.8432	0.8548	0.8678	0.8653	0.8757 - 3	0.8788 - 5
swin_s	0.8316	0.8554	0.8615	0.8667	0.8631	0.8752 - 4	0.8758 - 5
swin_t	0.8143	0.8588	0.8668	0.8683	0.8612	0.8750 - 4	0.8751 - 5
swin_v2_b	0.8412	0.8516	0.8604	0.8672	0.8645	0.8762 - 4	0.8766 - 5
swin_v2_s	0.8365	0.8592	0.8680	0.8733	0.8703	0.8805 - 4	0.8813 - 5
swin_v2_t	0.8204	0.8593	0.8683	0.8696	0.8647	0.8769 - 4	0.8772 - 5
vgg11	0.6909	0.8609	0.8602	0.8612	0.8410	0.8611 - 0	0.8480 - 0

vgg11_bn	0.7034	0.8626	0.8617	0.8627	0.8444	0.8626 - 0	0.8491 - 0
vgg13	0.6988	0.8619	0.8613	0.8622	0.8442	0.8621 - 0	0.8502 - 0
vgg13_bn	0.7151	0.8634	0.8623	0.8635	0.8451	0.8634 - 0	0.8522 - 0
vgg16	0.7156	0.8661	0.8652	0.8662	0.8478	0.8661 - 0	0.8556 - 0
vgg16_bn	0.7335	0.8679	0.8665	0.8680	0.8525	0.8679 - 0	0.8591 - 0
vgg19	0.7232	0.8656	0.8647	0.8656	0.8479	0.8656 - 0	0.8565 - 0
vgg19_bn	0.7420	0.8654	0.8641	0.8654	0.8510	0.8654 - 0	0.8560 - 0
vit_b_16	0.8102	0.8559	0.8637	0.8680	0.8621	0.8756 - 4	0.8762 - 5
vit_b_32	0.7592	0.8559	0.8631	0.8623	0.8556	0.8742 - 4	0.8743 - 5
vit_h_14	0.8848	0.8754	0.8777	0.8816	0.8791	0.8848 - 4	0.8850 - 6
vit_l_16	0.7969	0.8587	0.8618	0.8642	0.8603	0.8767 - 4	0.8767 - 4
vit_l_32	0.7696	0.8542	0.8591	0.8631	0.8561	0.8760 - 4	0.8750 - 5
wide_resnet101_2	0.8246	0.8384	0.8592	0.8691	0.8665	0.8787 - 5	0.8790 - 5
wide_resnet50_2	0.8155	0.7920	0.8524	0.8646	0.8611	0.8749 - 4	0.8752 - 6
efficientnetv2_xl	0.8555	0.7732	0.8109	0.8587	0.8604	0.8708 - 4	0.8740 - 6
vit_l_16_384	0.8702	0.8857	0.8855	0.8862	0.8801	0.8861 - 0	0.8837 - 0
vit_b_16_sam	0.8016	0.8825	0.8827	0.8833	0.8770	0.8832 - 0	0.8825 - 0
vit_b_32_sam	0.7368	0.8754	0.8758	0.8762	0.8661	0.8761 - 0	0.8755 - 0