Multi-Head Multi-Loss Model Calibration

Adrian Galdran[1,2,4(✉)], Johan W. Verjans[2,4], Gustavo Carneiro[2,4], and Miguel A. González Ballester[1,3,4]

[1] BCN Medtech, Universitat Pompeu Fabra, Barcelona, Spain
{adrian.galdran,ma.gonzalez}@upf.edu
[2] AIML, University of Adelaide, Adelaide, Australia
johan.verjans@adelaide.edu, g.carneiro@surrey.ac.uk
[3] University of Surrey, Guildford, UK
[4] Catalan Institution for Research and Advanced Studies (ICREA), Barcelona, Spain

Abstract. Delivering meaningful uncertainty estimates is essential for a successful deployment of machine learning models in the clinical practice. A central aspect of uncertainty quantification is the ability of a model to return predictions that are well-aligned with the actual probability of the model being correct, also known as model calibration. Although many methods have been proposed to improve calibration, no technique can match the simple, but expensive approach of training an ensemble of deep neural networks. In this paper we introduce a form of simplified ensembling that bypasses the costly training and inference of deep ensembles, yet it keeps its calibration capabilities. The idea is to replace the common linear classifier at the end of a network by a set of heads that are supervised with different loss functions to enforce diversity on their predictions. Specifically, each head is trained to minimize a weighted Cross-Entropy loss, but the weights are different among the different branches. We show that the resulting averaged predictions can achieve excellent calibration without sacrificing accuracy in two challenging datasets for histopathological and endoscopic image classification. Our experiments indicate that Multi-Head Multi-Loss classifiers are inherently well-calibrated, outperforming other recent calibration techniques and even challenging Deep Ensembles' performance. Code to reproduce our experiments can be found at https://github.com/agaldran/mhml_calibration.

Keywords: Model Calibration · Uncertainty Quantification

1 Introduction and Related Work

When training supervised computer vision models, we typically focus on improving their predictive performance, yet equally important for safety-critical tasks is their ability to express meaningful uncertainties about their own predictions

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-43898-1_11.

H. Greenspan et al. (Eds.): MICCAI 2023, LNCS 14222, pp. 108–117, 2023.
https://doi.org/10.1007/978-3-031-43898-1_11

[4]. In the context of machine learning, we often distinguish two types of uncertainties: *epistemic* and *aleatoric* [13]. Briefly speaking, epistemic uncertainty arises from imperfect knowledge of the model about the problem it is trained to solve, whereas aleatoric uncertainty describes ignorance regarding the data used for learning and making predictions. For example, if a classifier has learned to predict the presence of cancerous tissue on a colon histopathology, and it is tasked with making a prediction on a breast biopsy it may display epistemic uncertainty, as it was never trained for this problem [21]. Nonetheless, if we ask the model about a colon biopsy with ambiguous visual content, *i.e.* a hard-to-diagnose image, then it could express aleatoric uncertainty, as it may not know how to solve the problem, but the ambiguity comes from the data. This distinction between epistemic and aleatoric is often blurry, because the presence of one of them does not imply the absence of the other [12]. Also, under strong epistemic uncertainty, aleatoric uncertainty estimates can become unreliable [31].

Producing good uncertainty estimates can be useful, *e.g.* to identify test samples where the model predicts with little confidence and which should be reviewed [1]. A straightforward way to report uncertainty estimates is by interpreting the output of a model (maximum of its softmax probabilities) as its predictive confidence. When this confidence aligns with the actual accuracy we say that the model is calibrated [8]. Model calibration has been studied for a long time, with roots going back to the weather forecasting field [3]. Initially applied mostly for binary classification systems [7], the realization that modern neural networks tend to predict over-confidently [10] has led to a surge of interest in recent years [8]. Broadly speaking, one can attempt to promote calibration during training, by means of a post-processing stage, or by model ensembling.

Training-Time Calibration. Popular training-time approaches consist of reducing the predictive entropy by means of regularization [11], *e.g.* Label Smoothing [25] or MixUp [30], or loss functions that smooth predictions [26]. These techniques often rely on correctly tuning a hyper-parameter controlling the trade-off between discrimination ability and confidence, and can easily achieve better calibration at the expense of decreasing predictive performance [22]. Examples of medical image analysis works adopting this approach are Difference between Confidence and Accuracy regularization [20] for medical image diagnosis, or Spatially-Varying and Margin-Based Label Smoothing [14,27], which extend and improve Label Smoothing for biomedical image segmentation tasks.

Post-Hoc Calibration. Post-hoc calibration techniques like Temperature Scaling [10] and its variants [6,15] have been proposed to correct over or under-confident predictions by applying simple monotone mappings (fitted on a held-out subset of the training data) on the output probabilities of the model. Their greatest shortcoming is the dependence on the *i.i.d.* assumption implicitly made when using validation data to learn the mapping: these approaches suffer to generalize to unseen data [28]. Other than that, these techniques can be combined with training-time methods and return compounded performance improvements.

Model Ensembling. A third approach to improve calibration is to aggregate the output of several models, which are trained beforehand so that they have some diversity in their predictions [5]. In deep learning, model ensembles are considered to be the most successful method to generate meaningful uncertainty estimates [16]. An obvious weakness of deep ensembles is the requirement of training and then keeping for inference purposes a set of models, which results in a computational overhead that can be considerable for larger architectures. Examples of applying ensembling in medical image computing include [17,24].

In this work we achieve model calibration by means of multi-head models trained with diverse loss functions. In this sense, our approach is closest to some recent works on multi-output architectures like [21], where a multi-branch CNN is trained on histopathological data, enforcing specialization of the different heads by backpropagating gradients through branches with the lowest loss. Compared to our approach, ensuring correct gradient flow to avoid dead heads requires ad-hoc computational tricks [21]; in addition, no analysis on model calibration on in-domain data or aleatoric uncertainty was developed, focusing instead on anomaly detection. Our main **contribution** is a multi-head model that **I)** exploits multi-loss diversity to achieve greater confidence calibration than other learning-based methods, while **II)** avoiding the use of training data to learn post-processing mappings as most post-hoc calibration methods do, and **III)** sidesteping the computation overhead of deep ensembles.

2 Calibrated Multi-Head Models

In this section we formally introduce multi-head models [19], and justify the need for enforcing diversity on them. Detailed derivations of all the results below are provided in the online supplementary materials.

2.1 Multi-Head Ensemble Diversity

Consider a K-class classification problem, and a neural network U_θ taking an image $\mathbf{x}$ and mapping it onto a representation $U_\theta(\mathbf{x}) \in \mathbb{R}^{\mathbb{N}}$, which is linearly transformed by f into a logits vector $\mathbf{z} = f(U_\theta(\mathbf{x})) \in \mathbb{R}^{\mathbb{K}}$. This is then mapped into a vector of probabilities $\mathbf{p} \in [0,1]^K$ by a softmax operation $\mathbf{p} = \sigma(\mathbf{z})$, where $p_j = e^{z_j} / \sum_i e^{z_i}$. If the label of $\mathbf{x}$ was $y \in \{1, ..., K\}$, we can measure the error associated to prediction $\mathbf{p}$ with the cross-entropy loss $\mathcal{L}_{\text{CE}}(\mathbf{p}, y) = -\log(p_y)$.

We now wish to implement a multi-head ensemble model like the one shown in Fig. 1. For this, we replace f by M different branches $f^1, ..., f^M$, each of them still taking the same input but mapping it to different logits $\mathbf{z}^m = f^m(U_\theta(\mathbf{x}))$. The resulting probability vectors $\mathbf{p}^m = \sigma(\mathbf{z}^m)$ are then averaged to obtain a final prediction $\mathbf{p}^\mu = (1/M)\sum_m \mathbf{p}^m$. We are interested in backpropagating the loss $\mathcal{L}_{\text{CE}}(\mathbf{p}^\mu, y) = -\log(p^\mu_y)$ to find the gradient at each branch, $\nabla_{\mathbf{z}^m}\mathcal{L}_{\text{CE}}(\mathbf{p}^\mu, y)$.

Property 1: For the M-head classifier in Fig. 1, the derivative of the cross-entropy loss at head f^m with respect to $\mathbf{z}^m$ is given by

$$\nabla_{\mathbf{z}^m}\mathcal{L}_{\text{CE}}(\mathbf{p}^\mu, y) = \frac{p^m_y}{\sum_i p^i_y}(\mathbf{p}^\mu - \mathbf{y}), \tag{1}$$

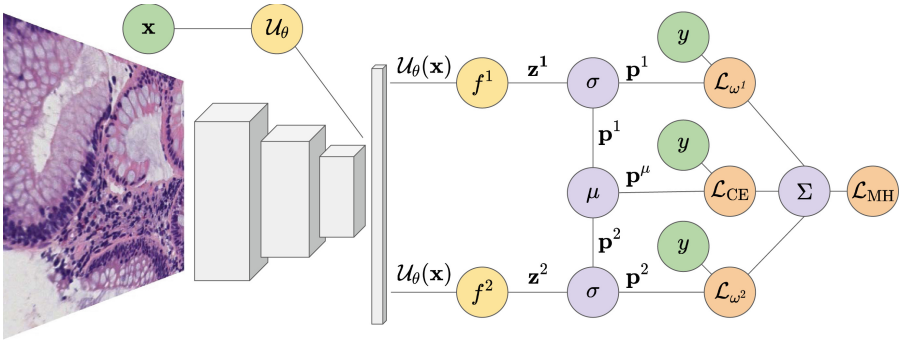

Fig. 1. A multi-head multi-loss model with M=2 heads. An image $\mathbf{x}$ goes through a neural network $\mathcal{U}_\theta$ and then is linearly transformed by M heads $\{f^m\}_{m=1}^M$, followed by softmax operations σ, into probability vectors $\{\mathbf{p}^m\}_{m=1}^M$. The final loss $\mathcal{L}_{\text{MH}}$ is the sum of per-head weighted-CE losses $\mathcal{L}_{\boldsymbol{\omega}^m\text{-CE}}(\mathbf{p}^m, y)$ and the CE loss $\mathcal{L}_{\text{CE}}(\mathbf{p}^\mu, y)$ of the average prediction $\mathbf{p}^\mu = \mu(\mathbf{p}^1, ..., \mathbf{p}^m)$. We modify the weights $\boldsymbol{\omega}^m$ between branches to achieve more diverse gradients during training.

where $\mathbf{y}$ is a one-hot representation of the label y.

From Eq. (1) we see that the gradient in branch m will be scaled depending on how much probability mass p_y^m is placed by f^m on the correct class relative to the total mass placed by all heads. In other words, if every head learned to produce a similar prediction (not necessarily correct) for a particular sample, then the optimization process of this network would result in the same updates for all of them. As a consequence, diversity in the predictions that make up the output $\mathbf{p}^\mu$ of the network would be damaged.

2.2 Multi-Head Multi Loss Models

In view of the above, one way to obtain more diverse gradient updates in a multi-head model during training could be to supervise each head with a different loss function. To this end, we will apply the weighted cross-entropy loss, given by $\mathcal{L}_{\boldsymbol{\omega}\text{-CE}}(\mathbf{p}, y) = -\omega_y \log(p_y^\mu)$, where $\boldsymbol{\omega} \in \mathbb{R}^K$ is a weight vector. In our case, we assign to each head a different weight vector $\boldsymbol{\omega}^m$ (as detailed below), in such a way that a different loss function $\mathcal{L}_{\boldsymbol{\omega}^m\text{-CE}}$ will supervise the intermediate output of each branch f^m, similar to deep supervision strategies [18] but enforcing diversity. The total loss of the complete model is the addition of the per-head losses and the overall loss acting on the average prediction:

$$\mathcal{L}_{\text{MH}}(\mathbf{p}, y) = \mathcal{L}_{\text{CE}}(\mathbf{p}^\mu, y) + \sum_{m=1}^{M} \mathcal{L}_{\boldsymbol{\omega}^m\text{-CE}}(\mathbf{p}^m, y), \tag{2}$$

where $\mathbf{p} = (\mathbf{p}^1, ..., \mathbf{p}^M)$ is an array collecting all the predictions the network makes. Since $\mathcal{L}_{\boldsymbol{\omega}\text{-CE}}$ results from just multiplying by a constant factor the conventional CE loss, we can readily calculate the gradient of $\mathcal{L}_{\text{MH}}$ at each branch.

Property 2: For the Multi-Loss Multi-Head classifier shown in Fig. 1, the gradient of the Multi-Head loss $\mathcal{L}_{\text{MH}}$ at branch f^m is given by:

$$\nabla_{\mathbf{z}^m} \mathcal{L}_{\text{MH}}(\mathbf{p}, y) = \left(\omega_y^m + \frac{p_y^m}{\sum_i p_y^i} \right) (\mathbf{p}^\mu - \mathbf{y}). \tag{3}$$

Note that having equal weight vectors in all branches fails to break the symmetry in the scenario of all heads making similar predictions. Indeed, if for any two given heads f^{m_i}, f^{m_j} we have $\boldsymbol{\omega}^{m_i} = \boldsymbol{\omega}^{m_j}$ and $\mathbf{p}^{m_i} \approx \mathbf{p}^{m_j}$, *i.e.* $\mathbf{p}^m \approx \mathbf{p}^\mu \ \forall m$, then the difference in norm of the gradients of two heads would be:

$$\|\nabla_{\mathbf{z}^{m_i}} \mathcal{L}_{\text{MH}}(\mathbf{p}, y) - \nabla_{\mathbf{z}^{m_j}} \mathcal{L}_{\text{MH}}(\mathbf{p}, y)\|_1 \approx |\omega_y^{m_i} - \omega_y^{m_j}| \cdot \|\mathbf{p}^\mu - \mathbf{y}\|_1 = 0. \tag{4}$$

It follows that we indeed require a different weight in each branch. In this work, we design a weighting scheme to enforce the specialization of each head into a particular subset of the categories $\{c_1, ..., c_K\}$ in the training set.

We first assume that the multi-head model has less branches than the number of classes in our problem, *i.e.* $M \leq K$, as otherwise we would need to have different branches specializing in the same category. In order to construct the weight vector $\boldsymbol{\omega}^m$, we associate to branch f^m a subset of N/K categories, randomly selected, for specialization, and these are weighed with $\omega_j^m = K$. Then, the remaining categories in $\boldsymbol{\omega}^m$ receive a weight of $\omega_j^m = 1/K$. For example, in a problem with 4 categories and 2 branches, we could have $\boldsymbol{\omega}^1 = [2, 1/2, 2, 1/2]$ and $\boldsymbol{\omega}^2 = [1/2, 2, 1/2, 2]$. If N is not divisible by K, the reminder categories are assigned for specialization to random branches.

2.3 Model Evaluation

When measuring model calibration, the standard approach relies on observing the test set accuracy at different confidence bands B. For example, taking all test samples that are predicted with a confidence around $c = 0.8$, a well-calibrated classifier would show an accuracy of approximately 80% in this test subset. This can be quantified by the Expected Calibration Error (ECE), given by:

$$\text{ECE} = \sum_{s=1}^{N} \frac{|B_s|}{N} |\text{acc}(B_s) - \text{conf}(B_s)|, \tag{5}$$

where $\bigcup_s B_s$ form a uniform partition of the unit interval, and $\text{acc}(B_s)$, $\text{conf}(B_s)$ are accuracy and average confidence (maximum softmax value) for test samples predicted with confidence in B_s.

In practice, the ECE alone is not a good measure in terms of practical usability, as one can have a perfectly ECE-calibrated model with no predictive power [29]. A binary classifier in a balanced dataset, randomly predicting always one class with $c = 0.5 + \epsilon$ confidence, has a perfect calibration and 50% accuracy. Proper Scoring Rules like Negative Log-Likelihood (NLL) or the Brier score are

alternative uncertainty quality metrics [9] that capture both discrimination ability and calibration: a model must be *both accurate and calibrated* to achieve a low PSR value. We report NLL, and also standard Accuracy, which contrary to ECE can be high even for badly-calibrated models. Finally, we show as summary metric the average rank when aggregating rankings of ECE, NLL, and accuracy.

3 Experimental Results

We now describe the data we used for experimentation, carefully analyze performance for each dataset, and end up with a discussion of our findings.

3.1 Datasets and Architectures

We conducted experiments on two datasets: **1)** the **Chaoyang** dataset[1], which contains colon histopathology images. It has 6,160 images unevenly distributed in 4 classes (29%, 19%, 37%, 15%), with some amount of label ambiguity, reflecting high aleatoric uncertainty. As a consequence, the best model in the original reference [32], applying specific techniques to deal with label noise, achieved an accuracy of 83.4%. **2) Kvasir**[2], a dataset for the task of endoscopic image classification. The annotated part of this dataset contains 10,662 images, and it represents a challenging classification problem due a high amount of classes (23) and highly imbalanced class frequencies [2]. For the sake of readability we do not show measures of dispersion, but we add them to the supplementary material (Appendix B), together with further experiments on other datasets.

We implement the proposed approach by optimizing several popular neural network architectures, namely a common ResNet50 and two more recent models: a ConvNeXt [23] and a Swin-Transformer [23]. All models are trained for 50 epochs, which was observed enough for convergence, using Stochastic Gradient Descent with a learning rate of $l = 1e\text{-}2$. Code to reproduce our results and hyperparameter specifications are shared at https://github.com/agaldran/mhml_calibration.

3.2 Performance Analysis

Notation: We train three different multi-head classifiers: 1) a 2-head model where each head optimizes for standard (unweighted) CE, referred to as **2HSL** (2 Heads-Single Loss); 2) a 2-head model but with each head minimizing a differently weighed CE loss as described in Sect. 2.2. We call this model **2HML** (2 Heads-Multi Loss)); 3) Finally, we increase the number of heads to four, and we refer to this model as **4HML**. For comparison, we include a standard single-loss one-head classifier (**SL1H**), plus models trained with Label Smoothing (**LS** [25]), Margin-based Label Smoothing (**MbLS** [22]), **MixUp** [30], and using the

[1] https://bupt-ai-cz.github.io/HSA-NRL/.
[2] https://datasets.simula.no/hyper-kvasir/.

DCA loss [20]. We also show the performance of Deep Ensembles (**D-Ens** [16]). We analyze the impact of Temperature Scaling [10] in Appendix A.

What we expect to see: Multi-Head Multi-Loss models should achieve a better calibration (low ECE) than other learning-based methods, ideally approaching Deep Ensembles calibration. We also expect to achieve good calibration without sacrificing predictive performance (high accuracy). Both goals would be reflected jointly by a low NLL value, and by a better aggregated ranking. Finally we would ideally observe improved performance as we increase the diversity (comparing **2HSL** to **2HML**) and as we add heads (comparing **2HML** to **4HML**).

Chaoyang: In Table 1 we report the results on the Chaoyang dataset. Overall, accuracy is relatively low, since this dataset is challenging due to label ambiguity, and therefore calibration analysis of aleatoric uncertainty becomes meaningful here. As expected, we see how Deep Ensembles are the most accurate method, also with the lowest NLL, for two out of the three considered networks. However, we also observe noticeable differences between other learning-based calibration techniques and multi-head architectures. Namely, all other calibration methods achieve lower ECE than the baseline (**SL1H**) model, but *at the cost of a reduced accuracy*. This is actually captured by NLL and rank, which become much higher for these approaches. In contrast, **4HML** achieves the second rank in two architectures, only behind Deep Ensembles when using a ResNet50 and a Swin-Transformer, and above any other **2HML** with a ConvNeXt, *even outperforming Deep Ensembles in this case*. Overall, we can see a pattern: multi-loss multi-head models appear to be extremely well-calibrated (low ECE and NLL values) without sacrificing accuracy, and as we diversify the losses and increase the number of heads we tend to improve calibration.

Table 1. Results on the **Chaoyang dataset** with different architectures and strategies. For each model, **best** and **second best** ranks are marked.

	ResNet50				ConvNeXt				Swin-Transformer			
	ACC$^{\uparrow}$	ECE$_{\downarrow}$	NLL$_{\downarrow}$	Rank$_{\downarrow}$	ACC$^{\uparrow}$	ECE$_{\downarrow}$	NLL$_{\downarrow}$	Rank$_{\downarrow}$	ACC$^{\uparrow}$	ECE$_{\downarrow}$	NLL$_{\downarrow}$	Rank$_{\downarrow}$
SL1H	80.71	5.79	53.46	6.0	81.91	6.94	50.98	6.3	83.09	8.73	52.75	5.0
LS	74.81	2.55	64.27	6.7	79.59	6.13	55.65	7.3	79.76	3.98	55.37	6.0
MbLS	75.02	3.26	63.86	6.7	79.53	2.94	53.44	5.3	80.24	5.06	54.18	5.7
MixUp	76.00	3.67	62.72	6.3	79.95	6.20	55.58	7.0	80.25	3.89	54.62	4.7
DCA	76.17	5.75	62.13	6.7	78.28	3.69	57.78	7.3	79.12	7.91	59.91	8.3
D-Ens	82.19	2.42	46.64	**1.0**	82.98	5.21	46.08	3.3	83.50	6.79	44.80	**2.7**
2HSL	80.97	4.36	51.42	4.0	81.94	4.30	46.71	4.3	82.90	8.20	54.19	5.7
2HML	80.28	4.49	51.86	5.3	81.97	3.66	45.96	**2.7**	82.79	5.01	46.12	3.7
4HML	81.13	3.09	49.44	**2.3**	82.17	1.79	44.73	**1.3**	82.89	4.80	46.70	**3.3**

Table 2. Results on the **Kvasir dataset** with different architectures and strategies. For each model, **best** and **second best** ranks are marked.

	ResNet50				ConvNeXt				Swin-Transformer			
	ACC$^{\uparrow}$	ECE$_{\downarrow}$	NLL$_{\downarrow}$	Rank$_{\downarrow}$	ACC$^{\uparrow}$	ECE$_{\downarrow}$	NLL$_{\downarrow}$	Rank$_{\downarrow}$	ACC$^{\uparrow}$	ECE$_{\downarrow}$	NLL$_{\downarrow}$	Rank$_{\downarrow}$
OneH	89.87	6.32	41.88	5.3	90.02	5.18	35.59	5.0	90.07	5.81	38.01	5.7
LS	88.13	14.63	53.96	7.7	88.24	6.97	42.09	6.7	88.74	9.20	43.46	8.7
MbLS	88.20	16.92	57.48	8.0	88.62	8.55	43.07	7.0	89.15	8.19	41.85	7.7
MixUp	87.60	10.28	50.69	7.3	87.58	8.96	48.88	8.7	89.23	2.11	35.52	4.3
DCA	87.14	3.84	40.50	6.0	85.27	4.11	46.78	7.3	87.62	4.38	38.44	7.3
D-Ens	90.76	3.83	32.09	2.3	90.76	3.34	29.74	3.0	90.53	3.94	29.36	3.3
2HSL	89.76	4.52	34.34	4.7	90.21	2.63	28.69	**2.7**	90.40	3.65	29.14	3.0
2HML	90.05	3.62	31.37	**2.0**	89.92	1.49	28.15	**2.7**	90.19	2.73	28.66	**2.7**
4HML	89.99	2.22	30.02	**1.7**	90.10	1.65	28.01	**2.0**	90.00	1.82	27.96	**2.3**

Kvasir: Next, we show in Table 2 results for the Kvasir dataset. Deep Ensembles again reach the highest accuracy and excellent calibration. Interestingly, methods that smooth labels (**LS**, **MbLS**, **MixUP**) show a strong degradation in calibration and their ECE is often twice the ECE of the baseline **SL1H** model. We attribute this to class imbalance and the large number of categories: smoothing labels might be ineffective in this scenario. Note that models minimizing the **DCA** loss do manage to bring the ECE down, although by giving up accuracy. In contrast, all multi-head models improve calibration while maintaining accuracy. Remarkably, **4HML** *obtains lower ECE than Deep Ensembles in all cases*. Also, for two out of the three architectures **4HML** ranks as the best method, and for the other one **2HML** reaches the best ranking.

4 Conclusion

Multi-Head Multi-Loss networks are classifiers with enhanced calibration and no degradation of predictive performance when compared to their single-head counterparts. This is achieved by simultaneously optimizing several output branches, each one minimizing a differently weighted Cross-Entropy loss. Weights are complementary, ensuring that each branch is rewarded for becoming specialized in a subset of the original data categories. Comprehensive experiments on two challenging datasets with three different neural networks show that Multi-Head Multi-Loss models consistently outperform other learning-based calibration techniques, matching and sometimes surpassing the calibration of Deep Ensembles.

Acknowledgments. This work was supported by a Marie Sk lodowska-Curie Fellowship (No 892297) and by Australian Research Council grants (DP180103232 and FT190100525).

References

1. Bernhardt, M., Ribeiro, F.D.S., Glocker, B.: Failure detection in medical image classification: a reality check and benchmarking testbed. Trans. Mach. Learn. Res. (2022)
2. Borgli, H., et al.: HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. Sci. Data **7**(1), 283 (2020). https://doi.org/10.1038/s41597-020-00622-y
3. Brier, G.W.: Verification of forecasts expressed in terms of probability. Mon. Weather Rev. **78**, 1 (1950)
4. Chua, M., et al.: Tackling prediction uncertainty in machine learning for healthcare. Nature Biomed. Eng., 1–8, December 2022. https://doi.org/10.1038/s41551-022-00988-x
5. Dietterich, T.G.: Ensemble methods in machine learning. In: Multiple Classifier Systems (2000). https://doi.org/10.1007/3-540-45014-9_1
6. Ding, Z., Han, X., Liu, P., Niethammer, M.: Local temperature scaling for probability calibration. In: ICCV (2021)
7. Ferrer, L.: Analysis and Comparison of Classification Metrics, September 2022. 10.48550/arXiv.2209.05355
8. Filho, T.S., Song, H., Perello-Nieto, M., Santos-Rodriguez, R., Kull, M., Flach, P.: Classifier Calibration: How to assess and improve predicted class probabilities: a survey, December 2021. 10.48550/arXiv.2112.10327
9. Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. J. Am. Stat. Assoc. **102**(477), 359–378 (2007). https://doi.org/10.1198/016214506000001437
10. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: ICML (2017)
11. Hebbalaguppe, R., Prakash, J., Madan, N., Arora, C.: A stitch in time saves nine: a train-time regularizing loss for improved neural network calibration. In: CVPR (2022)
12. Hüllermeier, E.: Quantifying Aleatoric and Epistemic Uncertainty in Machine Learning: Are Conditional Entropy and Mutual Information Appropriate Measures? September 2022. 10.48550/arXiv.2209.03302
13. Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Mach. Learn. **110**(3), 457–506 (2021). https://doi.org/10.1007/s10994-021-05946-3
14. Islam, M., Glocker, B.: Spatially varying label smoothing: capturing uncertainty from expert annotations. In: IPMI (2021). https://doi.org/10.1007/978-3-030-78191-0_52
15. Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., Flach, P.: Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In: NeurIPS (2019)
16. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: NeurIPS (2017)
17. Larrazabal, A.J., Martínez, C., Dolz, J., Ferrante, E.: Orthogonal ensemble networks for biomedical image segmentation. In: MICCAI (2021). https://doi.org/10.1007/978-3-030-87199-4_56
18. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: AISTATS (2015)

19. Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., Batra, D.: Why M Heads are Better than One: Training a Diverse Ensemble of Deep Networks, November 2015. https://doi.org/10.48550/arXiv.1511.06314
20. Liang, G., Zhang, Y., Wang, X., Jacobs, N.: Improved trainable calibration method for neural networks on medical imaging classification. In: British Machine Vision Conference (BMVC) (2020)
21. Linmans, J., Elfwing, S., van der Laak, J., Litjens, G.: Predictive uncertainty estimation for out-of-distribution detection in digital pathology. Med. Image Anal. (2023). https://doi.org/10.1016/j.media.2022.102655
22. Liu, B., Ben Ayed, I., Galdran, A., Dolz, J.: The devil is in the margin: margin-based label smoothing for network calibration. In: CVPR (2022)
23. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: ICCV, October 2021. https://doi.org/10.1109/ICCV48922.2021.00986
24. Ma, W., Chen, C., Zheng, S., Qin, J., Zhang, H., Dou, Q.: Test-time adaptation with calibration of medical image classification nets for label distribution shift. In: MICCAI (2022). https://doi.org/10.1007/978-3-031-16437-8_30
25. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: NeurIPS (2019)
26. Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P., Dokania, P.: Calibrating Deep Neural Networks using Focal Loss. In: NeurIPS (2020)
27. Murugesan, B., Liu, B., Galdran, A., Ayed, I.B., Dolz, J.: Calibrating Segmentation Networks with Margin-based Label Smoothing, September 2022. https://doi.org/10.48550/arXiv.2209.09641
28. Ovadia, Y., et al.: Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In: NeurIPS (2019)
29. Reinke, A., et al.: Understanding metric-related pitfalls in image analysis validation, February 2023. https://doi.org/10.48550/arXiv.2302.01790
30. Thulasidasan, S., Chennupati, G., Bilmes, J.A., Bhattacharya, T., Michalak, S.: On mixup training: improved calibration and predictive uncertainty for deep neural networks. In: NeurIPS (2019)
31. Valdenegro-Toro, M., Mori, D.S.: A deeper look into aleatoric and epistemic uncertainty disentanglement. In: CVPR Workshops (2022)
32. Zhu, C., Chen, W., Peng, T., Wang, Y., Jin, M.: Hard sample aware noise robust learning for histopathology image classification. IEEE Trans. Med. Imaging **41**(4), 881–894 (2022). https://doi.org/10.1109/TMI.2021.3125459