The Effects of Ensembling on Long-Tailed Data

E. Kelly Buchanan[1] **Geoff Pleiss**[2] **Yixin Wang**[3]

John P. Cunningham[1]

[1]Columbia University [2]University of British Columbia [3]University of Michigan

{ekb2154,jpc2181}@columbia.edu, geoff.pleiss@stat.ubc.ca, yixinw@umich.edu

Abstract

Deep ensembles are a popular approach to improve accuracy and calibration over single model performance [1], either by averaging logits [2–4], or probabilities [1, 5, 6] of multiple models. Recent theoretical work has shown that logit and probability ensembles have different effects on the model bias and variance [7, 8], but to our knowledge these benefits have not yet been used to inform how to create ensembles. In this work, we show that for balanced datasets, there is no significant difference between logit and probability ensembles in terms of accuracy and ranked calibration. In contrast, we show that in long-tailed datasets, there are gains from logit ensembling when combined with imbalance bias reduction losses. In turn, our results show that we can have consistent performance improvements using loss-aware ensembles when dealing with long-tail data.

1 Introduction

Ensembling, i.e. combining predictions of multiple models, is a standard approach to improve over single model performance [e.g., 1, 9–12]. Previous work has built ensembles by averaging either logits [2–4], or probabilities [5, 1, 6], selecting whichever approach gives better performance. Recent work has shown that logit and probability ensembles have different benefits [7, 13, 8]. Tassi et al. [13] suggests to use logit ensembles over probability ensembles especially if one cares about calibration. However, these conclusions were evaluated on small-scale models, using limited metrics [14] and only on balanced datasets. Gupta et al. [7] notes that we might prefer probability ensembling when there is significant disagreement across models, and that logit ensembling is more sensitive to extreme predictions. One scenario where we might expect models to disagree more is when there is data imbalance, where the tail classes have few training labels compared to the head classes. We note that in the imbalanced training literature, more work uses logit ensembling [15–19] as opposed to probability ensembling [20], although logit and probability ensembling are not directly compared.

Models deployed on imbalanced datasets typically employ re-weighting and re-sampling strategies [21–24]. When there is data imbalance at train time, but not at test time, there is an “imbalance" bias introduced due to the difference in the number of training samples across classes. Many current state-of-the-art approaches use the balanced softmax (BS) loss [25] and its extensions to address the imbalance bias [17–19, 26, 27]. For a recent review on long-tail learning, we refer the reader to Zhang et al. [24].

In this work, we want to answer whether probability and logit ensembling does make a large difference in practice, thus, our contributions proceed as follows:

- We perform a systematic comparison between logit and probability ensembling for a variety of models trained on balanced and imbalanced datasets.

Code is available at `https://github.com/ekellbuch/longtail_ensembles`

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

- We show little difference between logit and probability ensembling in balanced datasets, both in terms of accuracy and ranked calibration.
- We show that logit averaging is better when dealing with imbalanced data when combined with losses that target the imbalance bias (e.g [25]).
- We show that the data imbalance exacerbates the diversity across ensemble members for all classes which is higher than the diversity for the true class – and highest for ensembles of models trained with the balanced softmax loss.

Altogether, our results show that we can leverage the differences between logit and probability ensembling to improve model performance in the long-tail setting.

2 Setup

We focus on multi-class classification problems with inputs $\boldsymbol{x} \in \mathbb{R}^D$ and targets $y \in \{1, \dots, C\}$ where D is the number of features and C is the number of classes. The number of training data is $n = \sum_{k=1}^{C} n_k$, with n_k denoting the number of samples from class k. Furthermore, $\boldsymbol{\pi}$ is the vector of label frequencies, where $\pi_k = n_k/n$ is the label frequency of class k. A dataset is considered imbalanced when n_k differs across classes. In practice, the number of samples per class can decrease exponentially, and the tail classes can be heavy.

Deep Ensembles: A standard *deep ensemble* consists of M models $\boldsymbol{f}_1(\cdot), \dots, \boldsymbol{f}_M(\cdot)$ where each $\boldsymbol{f}_i$ maps $\boldsymbol{x}$ to the probability simplex in $\mathbb{R}^C$. Throughout the paper $M = 4$. We form ensembles by averaging model logits ($\boldsymbol{z}$) or probabilities:

$$\bar{\boldsymbol{f}}_{logit}(\boldsymbol{x}) \triangleq \text{softmax}\left(\tfrac{1}{M}\sum_{i=1}^{M} \boldsymbol{z}_i(\boldsymbol{x})\right) \qquad \bar{\boldsymbol{f}}_{prob}(\boldsymbol{x}) \triangleq \tfrac{1}{M}\sum_{i=1}^{M} \text{softmax}\left(\boldsymbol{z}_i(\boldsymbol{x})\right)$$

Experimental details: We form ensembles of models trained independently on balanced data and imbalanced data. The models trained on balanced data were obtained from [28, 29]. The models trained on imbalanced data were trained from scratch, following [25] using the loss functions described below. Then, the ensembles predictions are given by averaging the logits or probabilities of 4 models trained using the same loss and the same data.

Datasets and Models: Our experiments include models trained on balanced datasets (CIFAR10 and Imagenet), and on heavy-tailed datasets (CIFAR10-LT and CIFAR100 long-tail (LT)). **CIFAR10** [30]: We include 137 models from 32 different architectures trained on CIFAR10, each trained for 2-5 seeds [28]. Using these models, we form 207 ensembles which we evaluate on the test set of CIFAR10, and on the OOD datasets CIFAR 10.1 and CINIC10.

ImageNet [31] We include 78 "standard" models from Taori et al. [29], each corresponding to a different architecture. Using these models, we form 234 ensembles of the models trained on ImageNet which we evaluate on the test set of ImageNet, and the OOD datasets ImageNet V2MF, ImageNet-C Gaussian noise and Fog noise levels 1, 3, and 5.

CIFAR10-LT, and **CIFAR100-LT**: We train 5 seeds of ResNet 32 and ResNet-110 models on CIFAR10-LT and CIFAR100-LT datasets [21]. The CIFAR10/100 LT datasets are created following [23, 21], using a subset of the training set of CIFAR10/CIFAR100, where the number of samples per class is sampled according to an exponential function $n_i = n\mu^i$. Here i is the class index (0-indexed), where n is the original number of training images and $\mu^i = 0.5$.

Losses: Table 1 summarizes the losses used to train models [24]. We include the softmax cross entropy loss (CE), and common losses to handle imbalanced data by reweighted the loss function [25, 24]. We include the weighted softmax CE and d-weighted softmax CE losses which reweight the softmax loss by the sample frequencies, and which are set via the weight variable in the cross entropy loss function in pytorch. Unlike the weighted softmax CE loss, the d-weighted softmax CE ensures that the head classes are not up-weighted. Furthermore, we also include the balanced softmax loss [25], a loss which accommodates the label distribution shift, i.e. when the train test is imbalanced and the test set is balanced.

Temperature Scaling: We apply the temperature scaling loss [33] to the models trained on imbalanced data, to learn the temperature parameter T using the validation set. In the case of the imbalanced datasets, the validation set is created by sampling 10% of the samples in the train set that

Table 1: **Losses used to train models.** The loss $\mathcal{L}$ depends on the model outputs, where logits z and probabilities p. π_k is the label frequency of class $k \in [1 \dots C]$, y is the label and T is the temperature scaling parameter.

Loss	Formulation
Softmax CE (ERM)	$\mathcal{L}_{\text{ce}} = -\log(p_y)$
Weighted Softmax CE	$\mathcal{L}_{\text{wce}} = -\frac{1}{\pi_y}\log(p_y)$
d-Weighted Softmax CE	$\mathcal{L}_{\text{dwce}} = -\frac{1}{C\pi_y}\log(p_y)$
Balanced Softmax CE [25]	$\mathcal{L}_{\text{bs}} = -\log\left(\frac{\pi_y \exp(\boldsymbol{z}_y)}{\sum_j \pi_j \exp(\boldsymbol{z}_j)}\right)$
Temperature scaling [32]	$\mathcal{L}_{\text{ts}} = -\log\left(\frac{\exp(z_y/T)}{\sum_j \exp(z_j/T)}\right)$

are not used in CIFAR10-LT but exist in the train set of CIFAR10. The same process is followed for the models trained on CIFAR100-LT. Rahaman et al. [34] noted that the order in which we ensemble and apply temperature scaling can lead to different results, in particular in the low-data regime. Thus, to calibrate ensembles, we follow the pool-then-calibration approach [34]. In a nutshell, in pool-then calibrate, we first form the ensemble, and then fit a single temperature parameter T by minimizing a proper scoring rule (eg. cross-entropy) on the validation set.

Metrics: We compare the model performance using the accuracy, the per-class accuracy and the F1 score. The per-class accuracy and F1 score are better suited for imbalanced datasets. While the accuracy (0-1) is the ratio of correct predictions, the F1 score (0-1) is high when the correct predictions are not tampered by false alarms (high precision) and misses (high recall). We use four different metrics to compare the model calibration. Model calibration is generally compared using the negative log likelihood (NLL) or the Brier score (B) [35]:

$$\text{NLL}(\boldsymbol{f}(\boldsymbol{x}), y) \triangleq -\log\left(f^{(y)}(\boldsymbol{x})\right), \quad \text{B}(\boldsymbol{f}(\boldsymbol{x}), y) \triangleq \|\boldsymbol{f}(\boldsymbol{x}) - \mathbf{1}_y\|_2^2, \tag{1}$$

where $\mathbf{1}_y$ represents a one-hot encoding of y. However, both the NLL and Brier are sensitive to how the sharpness of the distribution of prediction probabilities of a model and thus can produce arbitrary rankings of different methods [14]. Thus, we compare the ensemble calibration using the Calibration area under the curve (AUC) [36], which measures the quality of the uncertainty estimates across a variety of decision thresholds. We include the calibration AUC of the Receiver Operation Characteristics (ROC) curve and the Precision Recall (PR) curve. The ROC curve measures the trade-off between correct predictions and incorrect prediction rates, and the PR curve measures the trade-off between precision and recall.

3 Experiments

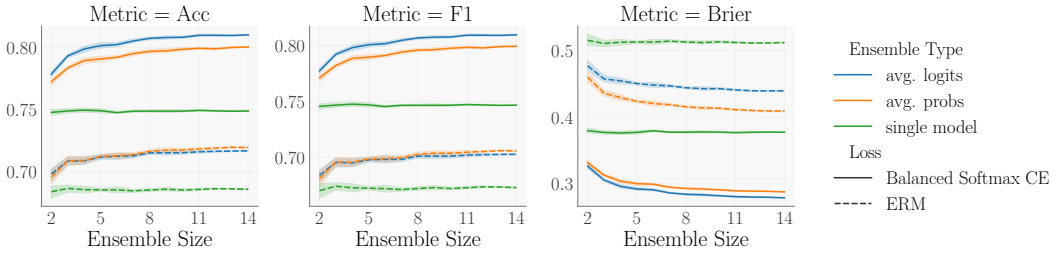

Figure 1: **Logit ensembles of models trained using a balanced softmax loss have the best performance regardless of the number of models in the ensemble.** We compare the performance of logit (blue) and probability (orange) ensembles of models trained via ERM and using a balanced softmax loss. Each ensemble is composed of ResNet32 models trained on the CIFAR10-LT dataset. The average single model (green) performance is included as a reference. The model weights are initialized using different random seeds.

Table 2 summarizes the performance of ResNet 32 models and ensembles trained on the imbalanced dataset CIFAR10-LT, using the losses in Table 1 when we fix the ensemble size to be $M = 4$. Table 2 shows that ensembling improves over single models across all metrics, regardless of the training loss (standard versus imbalanced) or the ensembling mechanism (logit versus probability averaging). However, upon closer inspection, we see a number of deviations.

Indeed, Table 2 shows that for the softmax loss and the re-weighted losses (weighted softmax and d-weighted softmax), probability or logit ensembling give similar results in terms of accuracy and F1 score (within error, and $< 1\%$ difference). In terms of calibration, probability averaging appears to be superior, as measured by Brier score and NLL, but this gap reduces when calibration is measured by the Calibration PR AUC.

Table 2: **Comparison of ResNet 32 models and ensembles trained on CIFAR10-LT**

Train Loss	Ensemble Type	Acc.	F1	Brier Score	NLL	Cal-PR AUC
Softmax CE (ERM)	single model	67.684 (± 0.249)	66.167 (± 0.338)	0.528 (± 0.004)	1.33 (± 0.011)	0.839 (± 0.002)
	avg. logits	69.928 (± 0.314)	68.332 (± 0.41)	0.471 (± 0.003)	1.119 (± 0.008)	0.874 (± 0.002)
	avg. probs	69.924 (± 0.286)	68.319 (± 0.389)	0.446 (± 0.003)	1.037 (± 0.009)	0.878 (± 0.002)
Balanced Softmax CE	single model	74.406 (± 0.178)	74.183 (± 0.185)	0.387 (± 0.002)	0.842 (± 0.004)	0.913 (± 0.0)
	avg. logits	**79.012 (± 0.186)**	**78.925 (± 0.206)**	**0.308 (± 0.002)**	**0.654 (± 0.003)**	**0.945 (± 0.0)**
	avg. probs	78.05 (± 0.085)	77.91 (± 0.087)	0.315 (± 0.002)	0.661 (± 0.003)	0.941 (± 0.0)
Weighted Softmax CE	single model	69.654 (± 0.122)	69.42 (± 0.138)	0.465 (± 0.002)	1.128 (± 0.005)	0.911 (± 0.001)
	avg. logits	72.16 (± 0.13)	71.931 (± 0.134)	0.411 (± 0.001)	0.95 (± 0.002)	0.931 (± 0.001)
	avg. probs	72.484 (± 0.099)	72.27 (± 0.106)	0.393 (± 0.001)	0.896 (± 0.002)	0.924 (± 0.001)
d-Weighted Softmax CE	single model	69.734 (± 0.239)	69.507 (± 0.27)	0.464 (± 0.003)	1.125 (± 0.009)	0.912 (± 0.001)
	avg. logits	72.556 (± 0.187)	72.382 (± 0.226)	0.41 (± 0.003)	0.949 (± 0.008)	0.931 (± 0.001)
	avg. probs	72.772 (± 0.181)	72.585 (± 0.222)	0.393 (± 0.003)	0.895 (± 0.006)	0.924 (± 0.001)

However, Table 2 shows that logit ensembling combined with the balanced softmax loss gives the best performance, across all metrics. Moreover, Fig. 1 shows that logit ensembling is superior to probability ensembling across increasing ensemble sizes when the ensemble members are trained to mitigate the bias introduced by data imbalance (i.e. trained with a balanced softmax). In Appx. A we show that our results give us state-of-the-art performance, not only compared to probability ensembles, but also when compared to implicit ensembles developed to mimic deep ensembles, and tailored to handle imbalanced datasets [20, 25, 37, 19]. Furthermore, in Appx. B, we show that these results hold for better calibrated models and ensembles.

In Appx. C we show that these results hold for additional imbalanced datasets across multiple architectures. Finally in Appx. D we show that these results hold for a variety of balanced datasets trained on ERM, where we see that the choice of ensembling mechanism makes little difference.

4 Discussion

We want to understand how ensembling interacts with the losses used for imbalanced data. In this section we show that we see better results using logit ensembling, as it does not change the imbalance debiasing effect of the balanced softmax loss, while probability averaging has an arbitrary effect in the (average single model) bias.

Gupta et al. [7] and Wood et al. [8] showed that the logit and probability ensemble cross entropy (or negative log likelihood for the true class) can be decomposed as:

$$\underbrace{-\mathop{\mathbb{E}}_{\mathcal{D}}\left[\boldsymbol{y}^T \cdot \ln \bar{\boldsymbol{q}}\right]}_{\text{logit ensemble NLL}} = \underbrace{-\frac{1}{M}\sum_{i=1}^{M} \boldsymbol{y}^T \cdot \ln \boldsymbol{q}_i{}^*}_{\text{average bias}} + \underbrace{\frac{1}{M}\sum_{i=1}^{M} \mathop{\mathbb{E}}_{\mathcal{D}}\left[D_{\mathrm{KL}}(\boldsymbol{q}_i^* || \boldsymbol{q}_i)\right]}_{\text{average variance}} - \underbrace{\mathop{\mathbb{E}}_{\mathcal{D}}\left[\frac{1}{M}\sum_{i=1}^{M} D_{\mathrm{KL}}(\bar{\boldsymbol{q}} || \boldsymbol{q}_i)\right]}_{\text{diversity}} \tag{2}$$

$$\underbrace{-\mathop{\mathbb{E}}_{\mathcal{D}}\left[\boldsymbol{y}^T \cdot \ln \boldsymbol{q}^\dagger\right]}_{\text{probability ensemble NLL}} = -\frac{1}{M}\sum_{i=1}^{M} \boldsymbol{y}^T \cdot \ln \boldsymbol{q}_i{}^* + \frac{1}{M}\sum_{i=1}^{M} \mathop{\mathbb{E}}_{\mathcal{D}}\left[D_{\mathrm{KL}}(\boldsymbol{q}_i^* || \boldsymbol{q}_i)\right]$$

$$- \underbrace{\mathop{\mathbb{E}}_{\mathcal{D}} \left[\sum_{i=1}^{M} \frac{1}{M} \left[\log \frac{1}{M} - \log \left[\frac{q_i^{(y)}}{\sum_{j=1}^{M} q_j^{(y)}} \right] \right] \right]}_{\text{dependency}}, \tag{3}$$

where $\mathcal{D}$ is a dataset sampled i.i.d. from $p(\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{q}_i$ is the probability output of model $i \in \{1 \dots M\}$, $\boldsymbol{y}$ is a one-hot encoding of y, $\boldsymbol{q}^\dagger$ is the probability ensemble (the arithmetic average of the ensemble member predictions), $\bar{\boldsymbol{q}}$ is the logit ensemble (the normalized geometric average of the ensemble member predictions), and $\boldsymbol{q}_i^* = \frac{1}{Z} \exp(\mathbb{E}_{\mathcal{D}}[\ln \boldsymbol{q}_{\mathcal{D}}])$ is the Bregman centroid. Moreover, $D_{\mathrm{KL}}(\boldsymbol{q} || \boldsymbol{q}')$ is an abuse of notation which represents the KL divergence between two categorical distributions with weights defined by the vectors $\boldsymbol{q}$ and $\boldsymbol{q}'$. Note that Eq. (2) is the same as Eq. 18 in [8] and Eq. (3) is the equation in Proposition 8 in [8] and is equivalent to Eq. 6 in [38].

Thus, the difference between the logit and probability ensemble NLL is the last term. For logit ensembles, the diversity is target ($\boldsymbol{y}$) independent but for probability ensembles, the diversity term is target dependent, so in this case we refer to this diversity term as a "dependency" term [8].

The "dependency" term in the probability ensemble NLL (Eq. (3)) can be interpreted as the KL divergence between two categorical distributions: the probability of sampling an ensemble member uniformly at random ($1/M$), and sampling an ensemble member proportional to its correct class prediction. Thus, the dependency will be large when the ensemble members predict different classes and zero when all ensemble members predict the correct class equally [38]. Meanwhile, the "diversity" term in the logit ensemble NLL is the average D_{KL} divergence between the ensemble predictions and the single model predictions across all classes. Thus, the diversity term will be large when the ensemble predictions are different and zero only if the ensemble members predictions are the same, for all classes. The qualitative similarity between these two behaviors suggest that the main difference between the "dependency" term and "diversity" term is the former's dependence on the true class label, whereas the latter is agnostic to the label.

We can subtract the logit ensemble NLL (Eq. (2)) from the probability ensemble NLL (Eq. (3)) to obtain:

$$\mathop{\mathbb{E}}_{\mathcal{D}} \left[\boldsymbol{y} \cdot \ln \bar{\boldsymbol{q}} - \boldsymbol{y} \cdot \ln \boldsymbol{q}^\dagger \right] = - \mathop{\mathbb{E}}_{\mathcal{D}} \left[\frac{1}{M} \sum_{i=1}^{M} D_{\mathrm{KL}}(\bar{\boldsymbol{q}} || \boldsymbol{q}_i) + \sum_{i=1}^{M} \frac{1}{M} \left[\log \frac{1}{M} - \log \left[\frac{q_i^{(y)}}{\sum_{j=1}^{M} q_j^{(y)}} \right] \right] \right]. \tag{4}$$

Eq. (4) tells us that logit ensembling is better than probability ensembling—in terms of the NLL—if and only if the diversity term is higher than the dependency term, i.e. the ensemble members are more likely to have different predictions for all classes, than different predictions for the true class (dependency term).

In the case of imbalanced datasets, we expect models trained via ERM to have high dependency and high diversity. We expect the diversity to be high, because we expect the models to provide different predictions across all classes given the imbalance in number of training samples for each class. Furthermore, we expect the dependency to be high, as we expect models to provide different predictions for the true classes, in particular for the tail classes, as shown in Fig. 2, which shows the per class pair-wise model disagreement across ensemble members. As we can see from Fig. 2, the models disagree more on tail classes, as expected.

Furthermore, Fig. 3 illustrates the logit ensemble diversity and the probability ensemble dependency term. Fig. 3 shows that ensembles of models trained with ERM have high diversity and dependency, and when we use any of the re-balancing losses, the dependency term reduces, i.e. the models are more likely to provide similar predictions for the true class. In turn, only for the balanced softmax loss is the ensemble diversity greater than the dependency, which, following Eq. (4), is necessary for logit ensembles to be superior to probability ensembles. To provide a rigorous underpinning for this observation, we complement our empirical results with a theoretical study in Appx. E, where we show that whenever we approximately minimize each loss (coupled with mild additional technical assumptions), it indeed holds that using the balanced softmax loss leads to the superiority of logit ensembling.

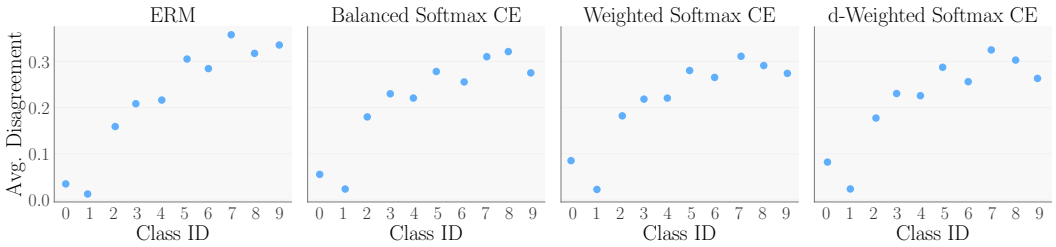

Figure 2: **Ensemble members disagree more in tail classes**. Per-class average disagreement for models in Table 2. The class ID is sorted from more to less training samples. Regardless of the imbalance loss (across columns), that ensembles disagree more for classes 5-9 and less for classes 0-4.

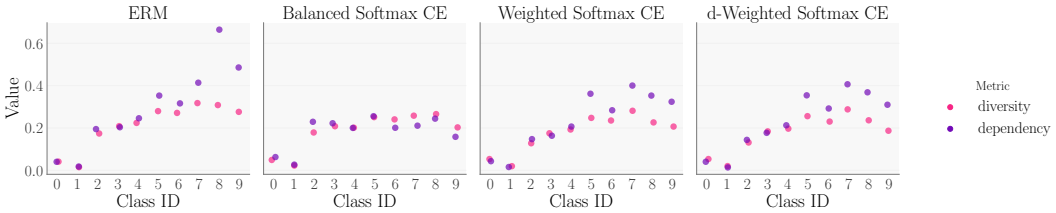

Figure 3: **The diversity is higher than the dependency only for the balanced softmax loss**. For each loss (column), we plot the logit ensemble diversity (from Eq. (2), magenta) and probability ensemble dependency (from Eq. (3), purple) terms for the models in Table 2.

5 Conclusion

Overall, our results show that for balanced datasets there is no significant difference between logit and probability ensembles in terms of accuracy and ranked calibration. However, we show that in imbalanced datasets, we can see gains from logit averaging when combined with bias reduction approaches. While our results focused on a balanced cross-entropy loss [25], we expect these results to hold for other losses that correct the bias introduced for ensembling, such as the logit adjustment loss [39], among others [18].

References

[1] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[3] Andrew Webb, Charles Reynolds, Wenlin Chen, Henry Reeve, Dan Iliescu, Mikel Lujan, and Gavin Brown. To ensemble or not ensemble: When does end-to-end training fail? In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 109–123. Springer, 2020.

[4] Raphael Gontijo-Lopes, Yann Dauphin, and Ekin Dogus Cubuk. No one representation to rule them all: Overlapping features of training methods. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=BK-4qbGgIE3`.

[5] Thomas G Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

[6] Ananya Kumar, Tengyu Ma, Percy Liang, and Aditi Raghunathan. Calibrated ensembles can mitigate accuracy tradeoffs under distribution shift. In *Uncertainty in Artificial Intelligence*, pages 1041–1051. PMLR, 2022.

[7] Neha Gupta, Jamie Smith, Ben Adlam, and Zelda E Mariet. Ensembles of classifiers: a bias-variance perspective. *Transactions of Machine Learning Research*, 2022.

[8] Danny Wood, Tingting Mu, Andrew Webb, Henry Reeve, Mikel Lujan, and Gavin Brown. A unified theory of diversity in ensemble learning. *arXiv preprint arXiv:2301.03962*, 2023.

[9] Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

[10] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[11] Dustin Tran, Jeremiah Liu, Du Phan Michael W Dusenberry, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, Neil Band, Tim GJ Rudner, Karan Singhal, Zachary Nado, Joost van Amersfoort, Andreas Kirsch, Rodolphe Jenatton, Nithum Thain, Honglin Yuan, Kelly Buchanan, Kevin Murphy, D Sculley, Yarin Gal, Zoubin Ghahramani, Jasper Snoek, and Balaji Lakshminarayanan. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

[12] E Kelly Buchanan, Michael W Dusenberry, Jie Ren, Kevin Patrick Murphy, Balaji Lakshminarayanan, and Dustin Tran. Reliability benchmarks for image segmentation. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

[13] Cedrique Rovile Njieutcheu Tassi, Jakob Gawlikowski, Auliya Unnisa Fitri, and Rudolph Triebel. The impact of averaging logits over probabilities on ensembles of neural networks. In *2022 Workshop on Artificial Intelligence Safety, AISafety 2022*, 2022.

[14] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

[15] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020.

[16] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 112–121, 2021.

[17] Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6949–6958, 2022.

[18] Emanuel Sanchez Aimar, Arvi Jonnarth, Michael Felsberg, and Marco Kuhlmann. Balanced product of calibrated experts for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19967–19977, 2023.

[19] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. *Advances in Neural Information Processing Systems*, 35:34077–34090, 2022.

[20] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020.

[21] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[22] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019.

[23] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[24] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[25] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

[26] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 715–724, 2021.

[27] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6908–6917, 2022.

[28] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, 2021.

[29] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems*, 2020.

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[32] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/guo17a.html`.

[33] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

[34] Rahul Rahaman et al. Uncertainty quantification and deep ensembles. *Advances in Neural Information Processing Systems*, 34:20063–20075, 2021.

[35] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[36] Ian D Kivlichan, Zi Lin, Jeremiah Liu, and Lucy Vasserman. Measuring and improving model-moderator collaboration using uncertainty estimation. *arXiv preprint arXiv:2107.04212*, 2021.

[37] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=D9I3drBz4UC`.

[38] Taiga Abe, E. Kelly Buchanan, Geoff Pleiss, and John Patrick Cunningham. The best deep ensembles sacrifice predictive diversity. In *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*, 2022. URL `https://openreview.net/forum?id=6sBiAIpkUiO`.

[39] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[40] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[41] Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew Mingbo Dai, and Dustin Tran. Training independent subnetworks for robust prediction. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=OGg9XnKxFAH`.

[42] Yeming Wen, Dustin Tran, and Jimmy Ba. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020.

A Implicit Ensembles

Given the computational costs of deep ensembles, several approaches have been proposed to form "implicit ensembles", which mimic ensemble performance with less computational requirements [40–42].

A variety of implicit ensembles have also been proposed to handle imbalanced datasets [20, 25, 37, 19]. However, as can be seen from Table 3, state-of-the-art implicit ensembles approaches still lag behind logit and probability ensembles. More importantly, Table 3 shows that logit ensembles of balanced softmax models give us the best performance.

Table 3: **Ensembles outperform most popular methods developed to handle imbalanced data** Comparison of different methods trained on the CIFAR100-LT dataset with an imbalance ratio of 100. * denotes number extracted from [19] and † denotes numbers extracted from Table 7.

Method	Accuracy
Softmax	$41.4^\dagger$
BBN [20]	44.7^*
Balanced Softmax [25]	46.1^*
RIDE [37]	48.0^*
SADE [19]	49.8^*
Logit Ensemble + Softmax	$44.68^\dagger$
Probability Ensemble + Softmax	$44.208^\dagger$
Probability Ensemble + Balanced Softmax	$51.8^\dagger$
Logit Ensemble + Balanced Softmax	$52^\dagger$

B Temperature scaling

Table 4 shows the performance from the better calibrated single models and ensembles. As stated in Sec. 2, the (average) single model performance is calculated after applying temperature scaling [33] to each model individually, and then averaging the individual model performances. Conversely, the ensemble performance (avg. logits and avg.probs in Table 4) is calculated after applying the pool-then-calibration approach from [34].

The results in Table 4 show that our conclusions also apply for ensembles with better calibration. First, the single model performance in Table 4 shows the calibration improvements achieved for each model on average. The results of the ensemble models in Table 4 show that even after we apply pool-then-calibrate, we not only get better performance, but using our proposed combination of logit ensembling models trained with a balanced softmax loss, we achieve the best performance. We expect these results to follow as we include other ensembling approaches, i.e. [6].

Table 4: **Logit ensembling + Balanced Softmax also gives the best performance after calibration.** Comparison of average single model and ensemble performance of ResNet 32 models trained on CIFAR10-LT before (Table 2)/after applying temperature scaling [33] to the individual models and pool-then-calbrate [34] to the ensembles.

Train Loss	Ensemble Type	Acc.	F1	Brier Score	NLL	Cal-PR auc
Softmax (ERM)	single model	67.684	66.167	0.528/0.476	1.33/1.099	0.839/0.843
	avg. logits (T=0.819)	69.928/74.450	68.332/73.885	0.471/0.374	1.119/0.821	0.874/0.914
	avg. probs (T=0.905)	69.924/74.390	68.319/73.797	0.446/0.372	1.037/0.803	0.878/0.914
Balanced Softmax CE	single model	74.406	74.183	0.387/0.372	0.842/0.821	0.913/0.915
	avg. logits (T=1.105)	79.012/82.260	78.925/82.225	0.308/0.259	0.654/0.555	0.945/0.962
	avg. probs (T=1.221)	78.05/81.550	77.91/81.513	0.315/0.266	0.661/0.558	0.941/0.960

C Comparing Ensembling Methods on Additional Imbalanced Datasets

Table 5 summarizes the models trained and evaluated on imbalanced datasets,. The results in this section are in line with conclusions from previous work [21, 25, 24]. In particular, these results show that models trained with re-weighting losses, such as the weighted softmax and a d-weighted softmax, have better performance than models trained using the standard cross-entropy when the number of classes is small (Table 2), but not when the number of classes is large (Table 7). Furthermore, these results show that models trained using a balanced softmax CE of outperform any other losses, regardless of the number of classes [24].

Table 5: **Ensemble loss comparison table guide**

Dataset	Data Type	Architecture	Table Number
CIFAR10-LT	InD	ResNet32	Table 2
	InD	ResNet110	Table 6
CIFAR100-LT	InD	ResNet32	Table 7
	InD	ResNet110	Table 8
CINIC-10 (eval only)	OOD	ResNet32	Table 9
	OOD	ResNet110	Table 10

Table 6: **Ensemble comparison of ResNet 110 models trained on CIFAR10-LT**

Train Loss	Ensemble Type	Acc.	F1	Brier Score	NLL	Cal-PR AUC
Softmax CE (ERM)	single model	71.16 (± 0.179)	70.351 (± 0.222)	0.492 (± 0.003)	1.347 (± 0.008)	0.865 (± 0.002)
	avg. logits	72.632 (± 0.165)	71.698 (± 0.196)	0.447 (± 0.003)	1.138 (± 0.009)	0.893 (± 0.002)
	avg. probs	72.914 (± 0.165)	72.011 (± 0.183)	0.414 (± 0.003)	1.018 (± 0.007)	0.895 (± 0.001)
Balanced Softmax CE	single model	76.004 (± 0.103)	75.837 (± 0.104)	0.381 (± 0.002)	0.871 (± 0.003)	0.928 (± 0.001)
	avg. logits	**79.652 (± 0.186)**	**79.611 (± 0.189)**	**0.31 (± 0.002)**	0.677 (± 0.005)	**0.951 (± 0.0)**
	avg. probs	78.876 (± 0.142)	78.786 (± 0.146)	**0.31 (± 0.002)**	**0.661 (± 0.003)**	0.947 (± 0.0)
Weighted Softmax CE	single model	70.754 (± 0.216)	70.535 (± 0.241)	0.474 (± 0.003)	1.244 (± 0.009)	0.911 (± 0.001)
	avg. logits	73.166 (± 0.259)	72.947 (± 0.286)	0.421 (± 0.003)	1.032 (± 0.01)	0.93 (± 0.001)
	avg. probs	73.446 (± 0.273)	73.257 (± 0.295)	0.394 (± 0.004)	0.944 (± 0.011)	0.925 (± 0.001)
d-Weighted Softmax CE	single model	70.348 (± 0.106)	70.114 (± 0.112)	0.481 (± 0.002)	1.268 (± 0.006)	0.908 (± 0.001)
	avg. logits	72.578 (± 0.122)	72.339 (± 0.129)	0.43 (± 0.002)	1.06 (± 0.005)	0.928 (± 0.001)
	avg. probs	72.802 (± 0.124)	72.569 (± 0.132)	0.402 (± 0.001)	0.968 (± 0.005)	0.923 (± 0.001)

Table 7: **Ensemble comparison of ResNet 32 models trained on CIFAR100-LT**

Train Loss	Ensemble Type	Acc.	F1	Brier Score	NLL	Cal-PR au c
Softmax (ERM)	single model	41.406 (± 0.014)	35.702 (± 0.037)	0.798 (± 0.001)	2.698 (± 0.008)	0.7 76 (± 0.002)
	avg. logits	44.68 (± 0.135)	38.814 (± 0.19)	0.746 (± 0.001)	2.41 (± 0.007)	0.805 (± 0.00 1)
	avg. probs	44.208 (± 0.107)	38.132 (± 0.107)	0.723 (± 0.001)	2.388 (± 0.006)	0.804 (± 0.001)
Balanced Softmax CE	single model	47.724 (± 0.171)	46.347 (± 0.233)	0.673 (± 0.002)	2.028 (± 0.004)	0.816 (± 0.001)
	avg. logits	**52.008 (± 0.159)**	**50.819 (± 0.197)**	0.612 (± 0.002)	**1.764 (± 0.004)**	**0.85 (± 0.002)**
	avg. probs	51.8 (± 0.186)	50.167 (± 0.246)	**0.61 (± 0.002)**	1.767 (± 0.004)	0.849 (± 0.001)
Weighted Softmax CE	single model	35.58 (± 0.426)	33.726 (± 0.422)	0.815 (± 0.003)	3.098 (± 0.017)	0.687 (± 0.005)
	avg. logits	39.682 (± 0.356)	37.432 (± 0.334)	0.761 (± 0.002)	2.786 (± 0.017)	0.734 (± 0.00 4)
	avg. probs	39.564 (± 0.413)	37.413 (± 0.393)	0.749 (± 0.002)	2.665 (± 0.015)	0.736 (± 0.004)
d-Weighted Softmax CE	single model	35.572 (± 0.235)	33.717 (± 0.213)	0.817 (± 0.002)	3.12 (± 0.014)	0.687 (± 0.004)
	avg. logits	39.572 (± 0.248)	37.267 (± 0.204)	0.763 (± 0.002)	2.801 (± 0.019)	0.736 (± 0.00 3)
	avg. probs	39.746 (± 0.267)	37.545 (± 0.281)	0.75 (± 0.002)	2.673 (± 0.014)	0.738 (± 0.002)

Table 8: **Ensemble comparison of ResNet 110 models trained on CIFAR100-LT**

Train Loss	Ensemble Type	Acc.	F1	Brier Score	NLL	Cal-PR a uc
Softmax (ERM)	single model	45.13 (± 0.19)	40.088 (± 0.245)	0.776 (± 0.001)	2.629 (± 0.008)	0. 804 (± 0.002)
	avg. logits	48.62 (± 0.154)	43.41 (± 0.203)	0.716 (± 0.001)	2.292 (± 0.006)	0.834 (± 0.001)
	avg. probs	48.062 (± 0.156)	42.638 (± 0.196)	0.686 (± 0.001)	2.239 (± 0.008)	0.833 (± 0.00 2)
Balanced Softmax CE	single model	50.09 (± 0.245)	49.006 (± 0.265)	0.665 (± 0.003)	2.039 (± 0.01)	0.832 (± 0.002)
	avg. logits	**55.548 (± 0.216)**	**54.602 (± 0.238)**	0.586 (± 0.002)	1.692 (± 0.006)	**0.868 (± 0.002)**
	avg. probs	55.308 (± 0.196)	53.929 (± 0.244)	**0.58 (± 0.002)**	**1.681 (± 0.006)**	0.866 (± 0.00 2)
Weighted Softmax CE	single model	36.186 (± 0.4)	34.204 (± 0.373)	0.809 (± 0.002)	3.14 (± 0.028)	0.709 (± 0.004)
	avg. logits	40.284 (± 0.303)	37.862 (± 0.292)	0.753 (± 0.002)	2.811 (± 0.031)	0.757 (± 0.002)
	avg. probs	40.406 (± 0.244)	38.118 (± 0.244)	0.74 (± 0.002)	2.672 (± 0.024)	0.755 (± 0.00 4)
d-Weighted Softmax CE	single model	35.728 (± 0.256)	33.841 (± 0.247)	0.816 (± 0.002)	3.201 (± 0.017)	0.699 (± 0.003)
	avg. logits	39.858 (± 0.235)	37.471 (± 0.216)	0.759 (± 0.003)	2.868 (± 0.023)	0.749 (± 0.002)
	avg. probs	39.832 (± 0.342)	37.576 (± 0.378)	0.746 (± 0.003)	2.721 (± 0.021)	0.748 (± 0.001)

Table 9: **Ensemble comparison of ResNet 32 models trained on CIFAR10-LT and evaluated on CINIC-10**

Train Loss	Ensemble Type	Acc.	F1	Brier Score	NLL	Cal-PR AUC
Softmax CE (ERM)	single model	57.94 (± 0.284)	55.225 (± 0.395)	0.689 (± 0.004)	1.797 (± 0.013)	0.757 (± 0.002)
	avg. logits	59.87 (± 0.449)	56.91 (± 0.552)	0.624 (± 0.005)	1.528 (± 0.012)	0.806 (± 0.002)
	avg. probs	60.01 (± 0.503)	57.14 (± 0.582)	0.587 (± 0.004)	1.431 (± 0.012)	0.811 (± 0.003)
Balanced Softmax CE	single model	63.35 (± 0.247)	62.768 (± 0.273)	0.548 (± 0.004)	1.231 (± 0.011)	0.829 (± 0.003)
	avg. logits	**68.45 (± 0.232)**	**67.972 (± 0.239)**	**0.457 (± 0.004)**	0.993 (± 0.011)	**0.876 (± 0.003)**
	avg. probs	67.32 (± 0.47)	66.639 (± 0.528)	0.458 (± 0.004)	**0.981 (± 0.01)**	0.873 (± 0.003)
Weighted Softmax CE	single model	57.77 (± 0.244)	56.868 (± 0.252)	0.649 (± 0.003)	1.683 (± 0.008)	0.808 (± 0.002)
	avg. logits	60.28 (± 0.553)	59.305 (± 0.499)	0.591 (± 0.004)	1.463 (± 0.009)	0.844 (± 0.002)
	avg. probs	60.73 (± 0.546)	59.758 (± 0.434)	0.56 (± 0.004)	1.367 (± 0.011)	0.842 (± 0.002)
d-Weighted Softmax CE	single model	57.99 (± 0.275)	57.098 (± 0.264)	0.649 (± 0.005)	1.669 (± 0.013)	0.812 (± 0.002)
	avg. logits	60.35 (± 0.389)	59.28 (± 0.396)	0.589 (± 0.004)	1.446 (± 0.011)	0.845 (± 0.001)
	avg. probs	60.9 (± 0.478)	59.906 (± 0.54)	0.561 (± 0.003)	1.364 (± 0.009)	0.839 (± 0.001)

Table 10: **Ensemble comparison of ResNet 110 models trained on CIFAR10-LT and evaluated on CINIC-10**

Train Loss	Ensemble Type	Acc.	F1	Brier Score	NLL	Cal-PR AUC
Softmax CE (ERM)	single model	60.8 (± 0.248)	59.139 (± 0.322)	0.665 (± 0.004)	1.878 (± 0.014)	0.788 (± 0.002)
	avg. logits	62.7 (± 0.411)	60.749 (± 0.544)	0.609 (± 0.006)	1.607 (± 0.016)	0.83 (± 0.003)
	avg. probs	63.1 (± 0.481)	61.24 (± 0.658)	0.565 (± 0.005)	1.46 (± 0.012)	0.835 (± 0.003)
Balanced Softmax CE	single model	65.2 (± 0.187)	64.542 (± 0.258)	0.554 (± 0.002)	1.301 (± 0.007)	0.85 (± 0.001)
	avg. logits	**68.56 (± 0.452)**	**68.118 (± 0.488)**	0.468 (± 0.005)	1.04 (± 0.01)	**0.887 (± 0.002)**
	avg. probs	67.52 (± 0.454)	66.866 (± 0.533)	**0.459 (± 0.003)**	**1.006 (± 0.008)**	0.882 (± 0.003)
Weighted Softmax CE	single model	58.2 (± 0.167)	57.121 (± 0.189)	0.678 (± 0.003)	1.881 (± 0.015)	0.812 (± 0.002)
	avg. logits	60.34 (± 0.219)	59.155 (± 0.237)	0.616 (± 0.005)	1.608 (± 0.016)	0.848 (± 0.002)
	avg. probs	60.77 (± 0.277)	59.591 (± 0.302)	0.575 (± 0.004)	1.484 (± 0.014)	0.843 (± 0.003)
d-Weighted Softmax CE	single model	58.78 (± 0.157)	57.808 (± 0.159)	0.674 (± 0.003)	1.885 (± 0.01)	0.818 (± 0.002)
	avg. logits	61.21 (± 0.344)	60.124 (± 0.413)	0.617 (± 0.004)	1.624 (± 0.009)	0.851 (± 0.001)
	avg. probs	61.32 (± 0.179)	60.311 (± 0.262)	0.574 (± 0.003)	1.487 (± 0.009)	0.847 (± 0.002)

D Comparison of Logit and Probability Ensembles on Balanced Datasets

Table 11 outlines the figure IDs of the plots comparing the performances of probability and logit ensembles trained and evaluated on a variety of balanced datasets using multiple metrics. Sec. 2 provides a detailed description of the datasets and metrics. Overall, this section shows that there are not significant differences between logit and probability ensembles trained and evaluated on balanced datasets.

Table 11: **Balanced Dataset comparisons.** Summary with the figure ID of the plots comparing the performance of probability and logit ensembles.

Train Dataset	Test Dataset	Data Type	Metrics	Figure
CIFAR10	CIFAR10	InD	0-1 error, F1 score	Fig. 4
			Brier Score, NLL, Calibration ROC/PR AUC	Fig. 5
CIFAR10	CINIC10	OOD	0-1 error, F1 score	Fig. 4
			Brier Score, NLL, Calibration ROC/PR AUC	Fig. 5
CIFAR10	CIFAR10.1	OOD	0-1 Error, F1 score	Fig. 6
			Brier Score, NLL, Calibration ROC/PR AUC	Fig. 7
ImageNet	ImageNet	InD	0-1 error, F1 score	Fig. 4
			Brier Score, NLL, Calibration ROC/PR AUC	Fig. 5
ImageNet	ImageNetV2MF	OOD	0-1 error, F1 score	Fig. 4
			Brier score, NLL, Calibration ROC/PR AUC	Fig. 5
ImageNet	ImageNet-C	OOD	0-1 error	Fig. 8
			F1 score	Fig. 9
			Brier Score	Fig. 10
			NLL	Fig. 11
			Calibration ROC AUC	Fig. 12
			Calibration PR AUC	Fig. 13

Logit and probability ensembling are no different in terms of model error for models trained on balanced datasets. Fig. 4 compares the ensemble performance of probability vs logit ensembles for a variety of balanced datasets trained via ERM. Fig. 4 shows that there is no significant difference between probability and logit ensembles in terms of the 0-1 error or F1 score for any dataset, and regardless of the level of model performance, i.e. ensembles with low 0-1 error or high 0-1 error.

Logit and probability ensembling are not significantly different in terms of ranked calibrations for models trained on balanced datasets. Fig. 5 illustrates the Brier score, NLL, Calibration ROC AUC and Calibration PR AUC for the same ensembles in Fig. 4. Fig. 5 shows some differences between the NLL and Brier score of logit and probability ensembles, but we note that comparing ensembles using the NLL or Brier score can produce an arbitrary rankings [14]. When comparing the ensemble calibration using the ranked calibration metrics, i.e. in terms of the Calibration ROC AUC and Calibration PR AUC [11], we see differences in terms of the Calibration ROC AUC in the range of ≤ 0.01. Meanwhile, when comparing the Calibration PR AUCs in the bottom row of Fig. 5: we can see that the differences between logit and probability ensembling vanish.

Fig. 7 illustrates the calibration metrics of the ensembles in Fig. 4, for the models trained on CIFAR10 and here evaluated on CIFAR10.1. Fig. 10 to Fig. 13 illustrate the calibration metrics of the ensembles in Fig. 4, for models trained on ImageNet and here evaluated on ImageNet-C. From these plots we can see that the differences between logit and probability ensembling reduce/vanish when comparing models in terms of Calibration PR AUC.

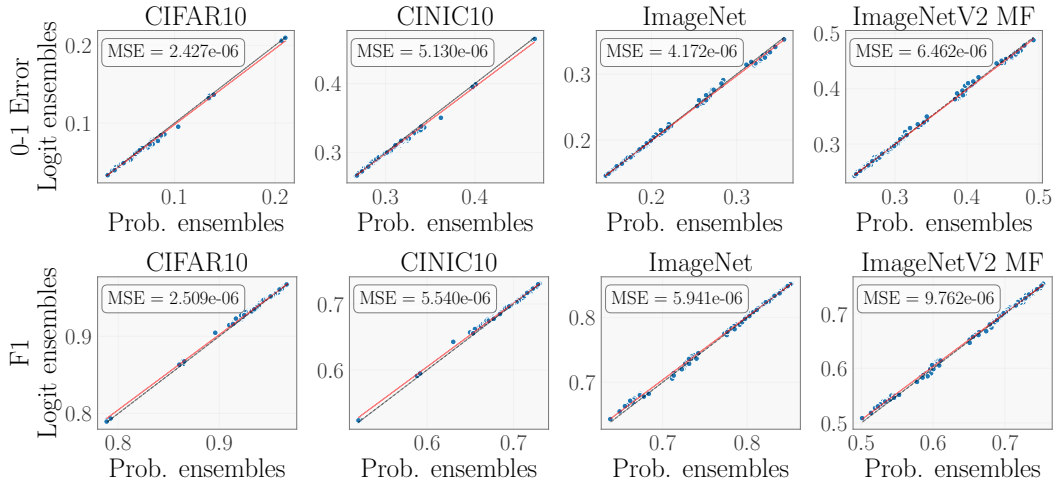

Figure 4: **Logit and probability ensembling have no significant differences for balanced datasets**. In each plot, each marker depicts the performance of an ensemble formed by averaging model probabilities (x-axis) vs averaging model logits (y-axis). The black line represents $(x = y)$, and the read line is the linear fit of (x, y). The box shows the mean squared error (MSE) between (x, y). The first two columns include 205 markers corresponding to ensembles of models trained on CIFAR10 and evaluated on the test set of CIFAR10 and CIFAR10.1 The last two columns include 234 markers corresponding to ensembles of models trained on ImageNet and evaluated on the test set of ImageNet and ImageNetV2 MF, as described in Sec. 2.

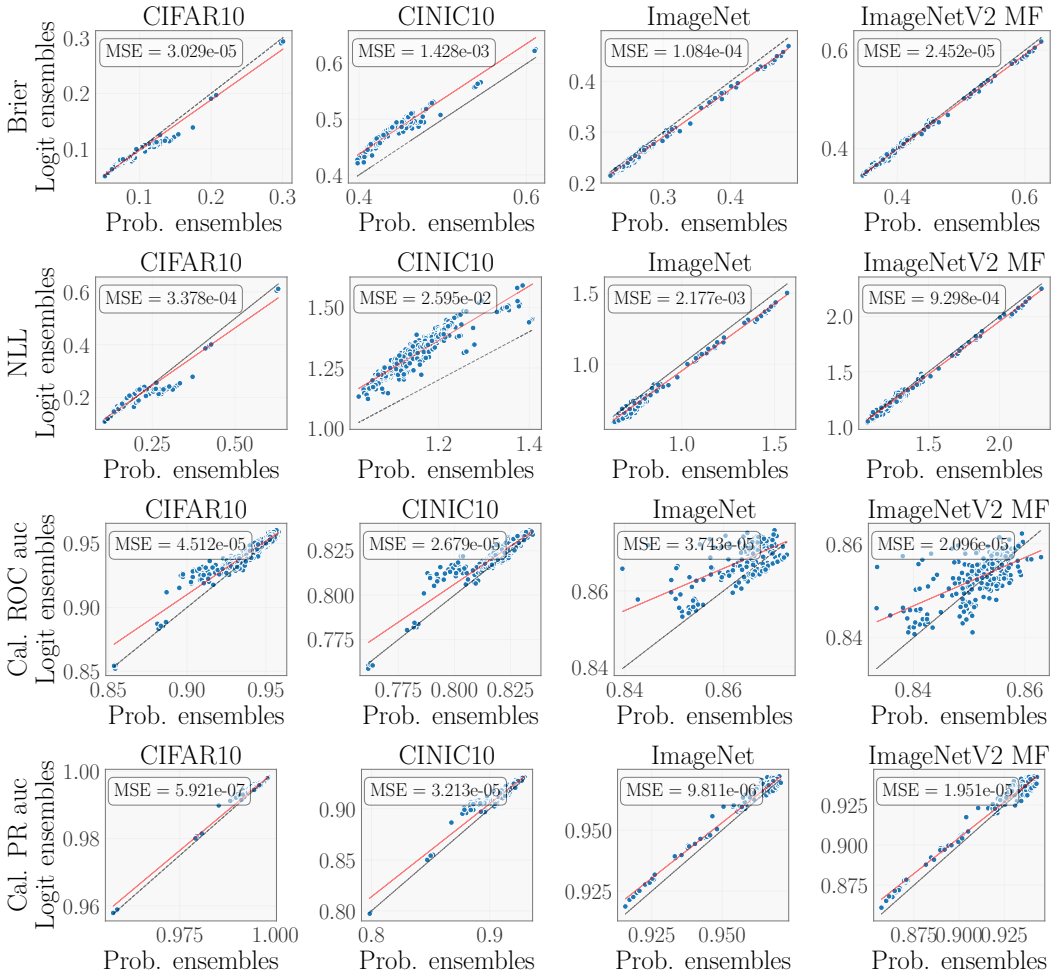

Figure 5: **Logit and probability ensembles have no major differences in terms of calibration**. Same conventions as Fig. 4 for a variety of calibration metrics.

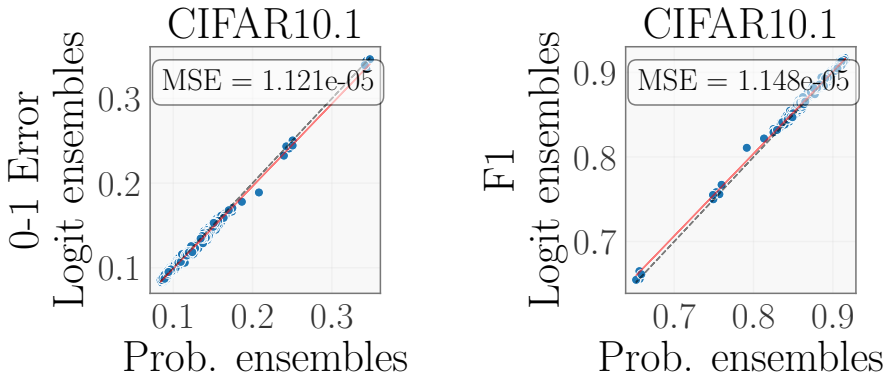

Figure 6: **Logit and probability ensembling are no different in terms of the 0-1 error**. Same conventions and conclusions as Fig. 4, with the ensembles formed from models trained on CIFAR10 are evaluated on CIFAR10.1.

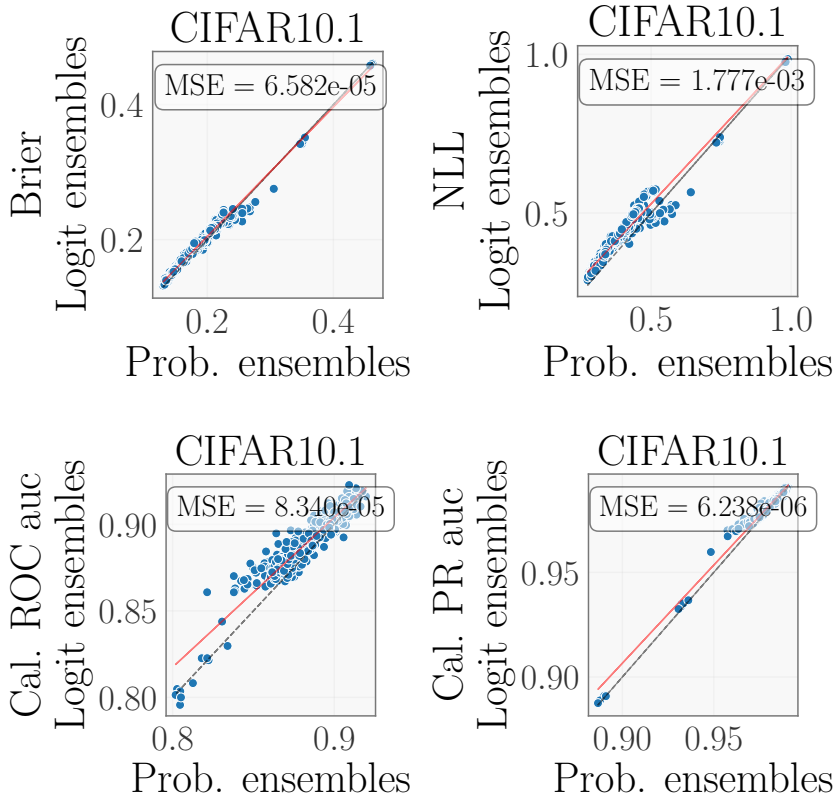

Figure 7: **Calibration metrics of probability vs logit ensembles of models trained on CIFAR10 and evaluated on CIFAR10.1**. Same conventions as Fig. 6 for a variety of calibration metrics.

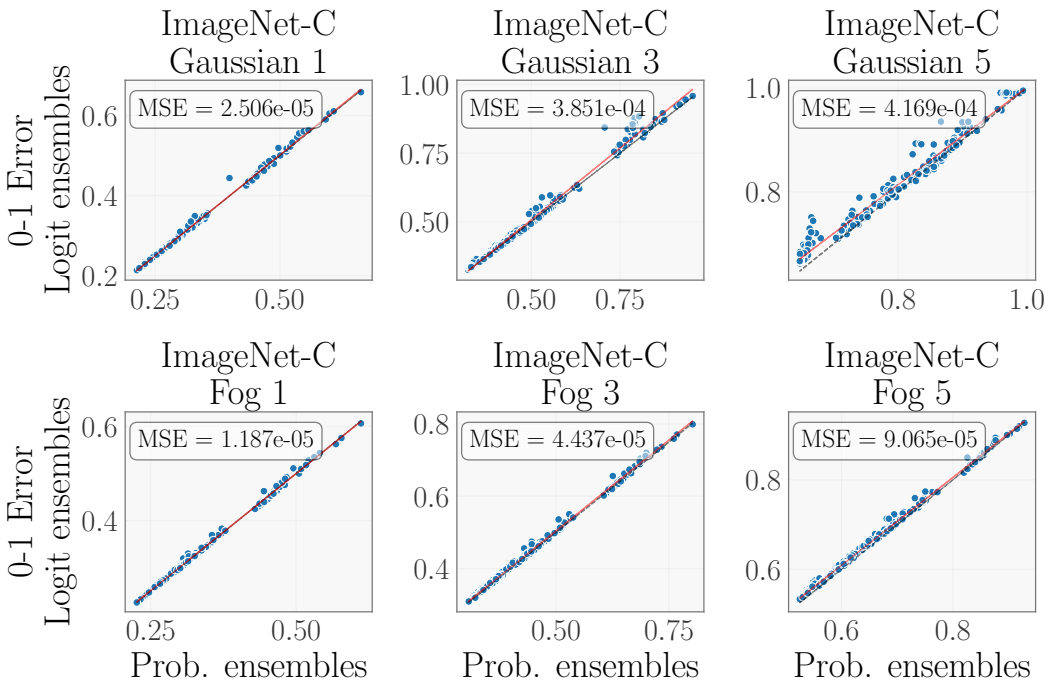

Figure 8: **Logit and probability ensembling have no significant differences in terms of 0-1 error**. Same conventions and conclusions as Fig. 4, where the ensembles formed from models trained on ImageNet are evaluated on ImageNet-C.

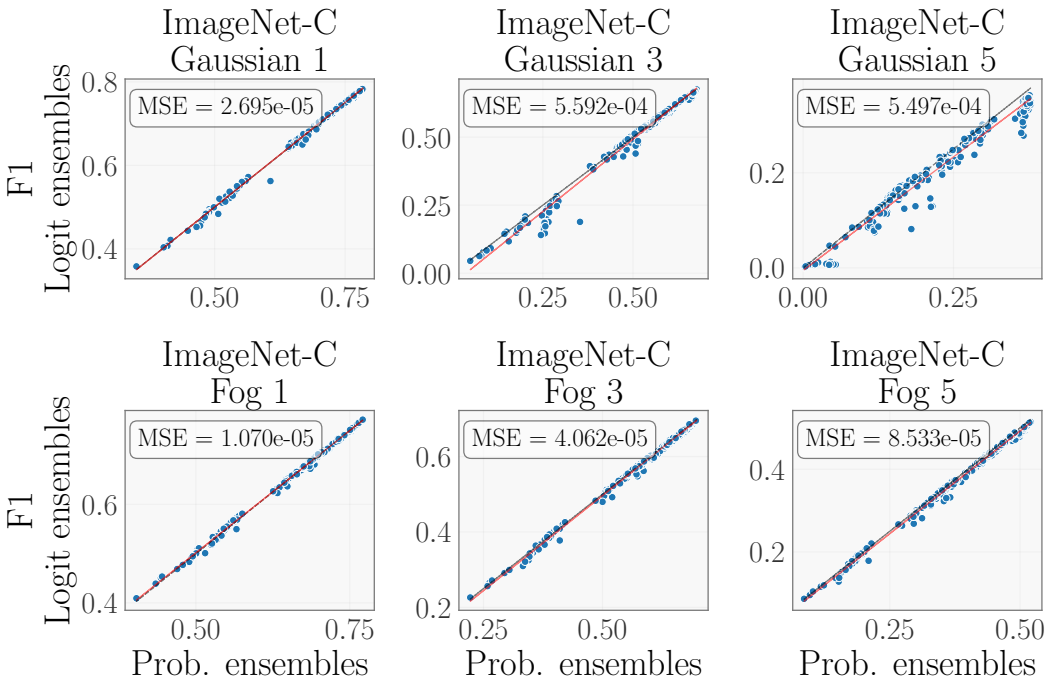

Figure 9: **Logit and probability averaging have no significant differences in terms of F1 score**. Same conventions and conclusions as Fig. 4, where the ensembles formed from models trained on ImageNet are evaluated on ImageNet-C.

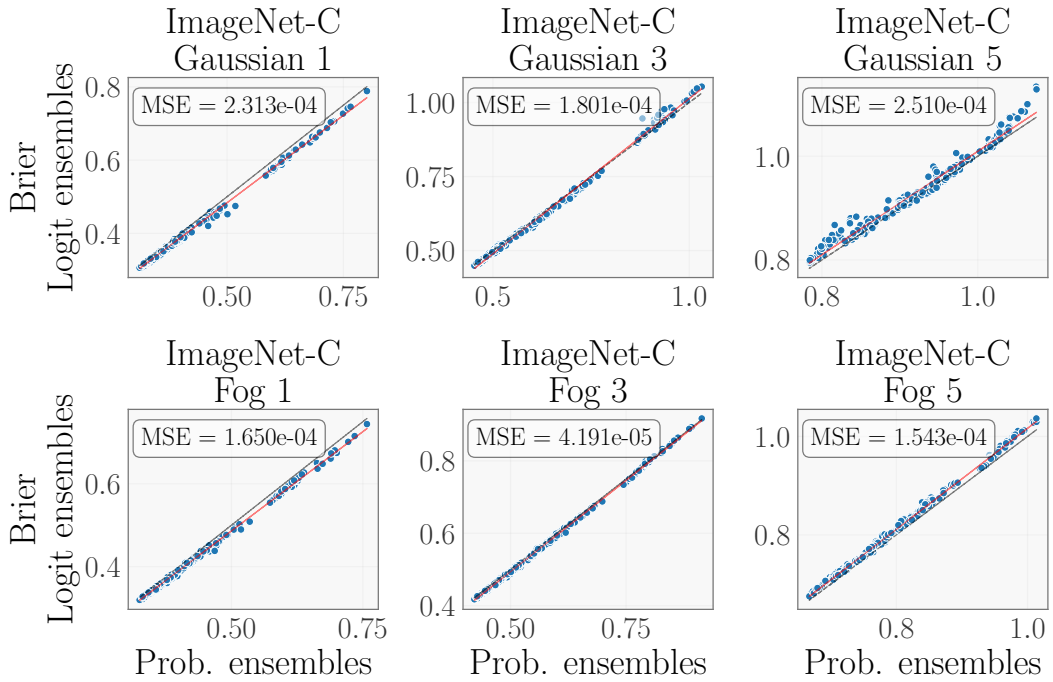

Figure 10: **Brier score of probability vs logit ensembles of models trained on ImageNet and evaluated on ImageNet-C.**

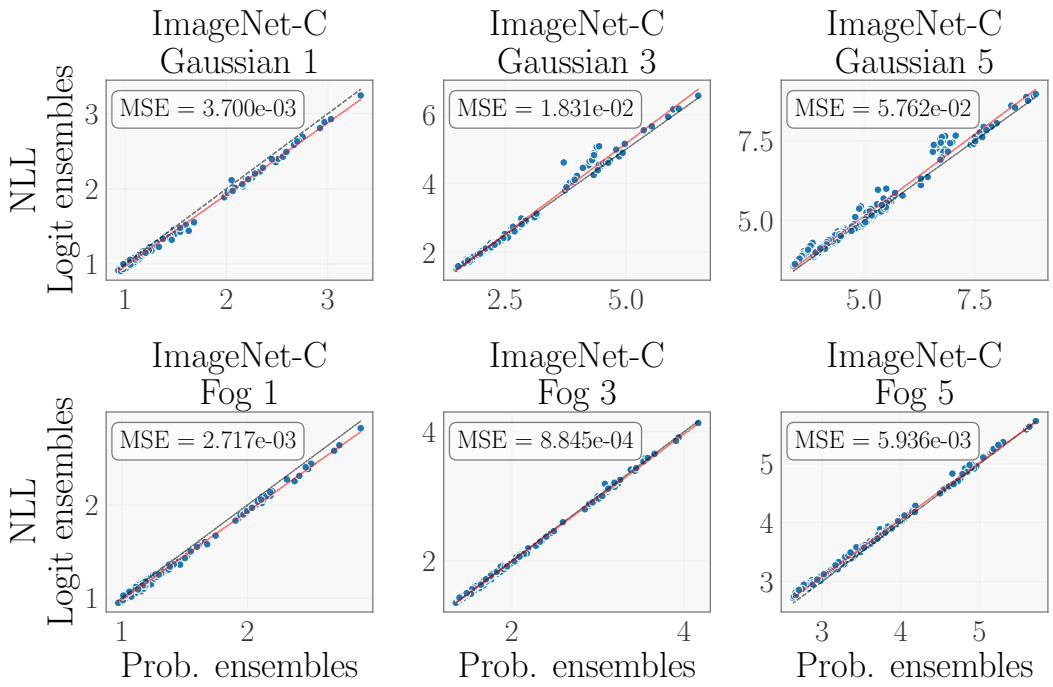

Figure 11: **NLL of probability vs logits ensembles of models trained on ImageNet and evaluated on ImageNet-C.**

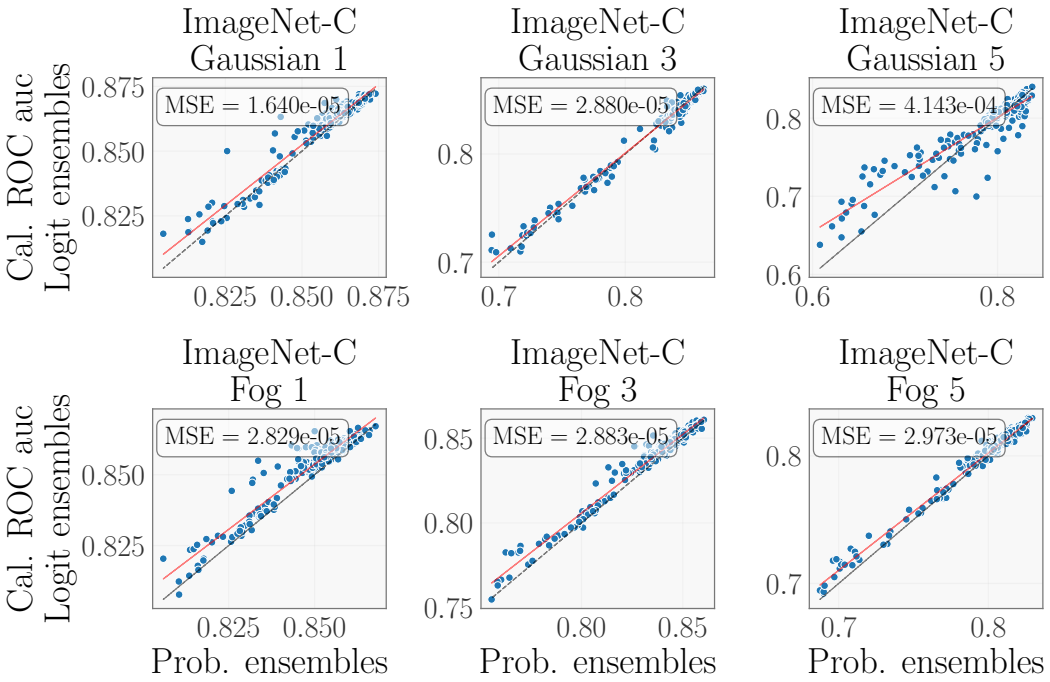

Figure 12: **Calibration ROC AUC of probability vs logit ensembles of models trained on ImageNet and evaluated on ImageNet-C**.

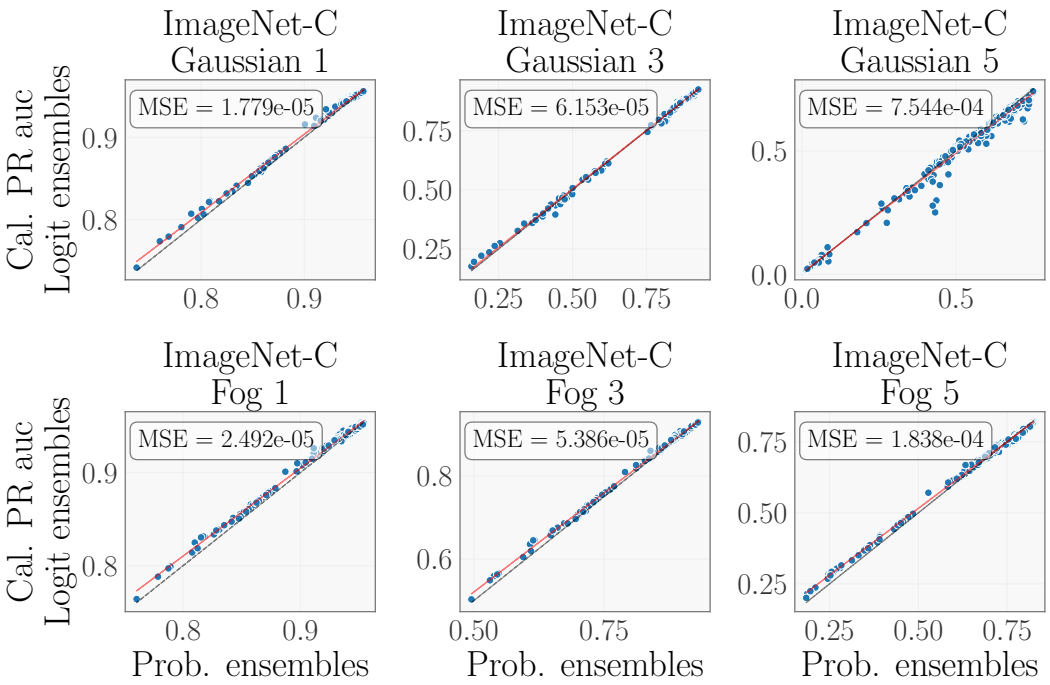

Figure 13: **Calibration PR AUC of probability vs logit ensembles of models trained on ImageNet and evaluated on ImageNet-C**.

E Theoretical arguments

ϵ-approximate optimal π-balanced loss. To consider the diversity of the ensemble members, we first consider ϵ-approximate optimal solutions, namely solutions that achieve a loss value at most ϵ away from the optimal loss values. As one independently optimize each ensemble members according to their loss, the ensemble members are likely uniform draws from the set of ϵ-approximate optimal solutions (for some small ϵ).

Lemma E.1 (Impact of balanced softmax on diversity)**.** *Denote $z^*(x)$ as the logit predictions that achieve the optimal K-class π-balanced softmax loss with weights $\pi = (\pi_1, \dots, \pi_K)$. The logit predictions $z(x)$ achieve the ϵ-approximate optimal π-balanced loss for all samples must satisfy*

$$\sum_{j=1}^{K} \pi_j \exp(z_j(x)) \left(\frac{\exp(z_y(x) - z_y^*(x))}{\exp(z_j(x) - z_j^*(x))} - \exp(\epsilon) \right) < 0, \qquad \forall (x, y). \tag{5}$$

Proof. We consider $z(x)$ that achieve ϵ-approximate optimal balanced softmax loss, i.e.

$$\left| \log \frac{\pi_y \exp(z_y(x))}{\sum_{j=1}^{K} \pi_j \exp(z_j(x)} - \sum_{(x,y)} \log \frac{\pi_y \exp(z_y^*(x))}{\sum_{j=1}^{K} \pi_j \exp(z_j^*(x)} \right| < \epsilon. \tag{6}$$

By re-arranging the terms, we have

$$\frac{\exp(z_y(x) - z_y^*(x))}{\sum_{j=1}^{K} \pi_j \exp(z_j(x)) / \sum_{j=1}^{K} \pi_j \exp(z_j^*(x))} < \exp(\epsilon), \tag{7}$$

$$\sum_{j=1}^{K} \pi_j \exp(z_j^*(x)) \left(\exp(z_y(x) - z_y^*(x)) - \exp(z_j(x) - z_j^*(x) + \epsilon) \right) < 0, \tag{8}$$

$$\sum_{j=1}^{K} \pi_j \exp(z_j(x)) \left(\frac{\exp(z_y(x) - z_y^*(x))}{\exp(z_j(x) - z_j^*(x))} - \exp(\epsilon) \right) < 0. \tag{9}$$

□

How diverse are ϵ-optima of classical and balanced softmax loss? We consider the range of $z(x)$ solutions that can satisfy Eq. (5) under different π. When $\pi_1 = \cdots = \pi_K = \frac{1}{K}$, the π-balanced softmax loss corresponds to the classical softmax loss. When $\pi_j = n_j/n, \forall j$, then the π-balanced softmax loss corresponds to Ren et al. [25].

For any solution $\hat{z}(x)$ that satisfies Eq. (5) under $\pi_1 = \cdots = \pi_K = \frac{1}{K}$, then any solution $\tilde{z}(x)$ that satisfies

$$\frac{\exp(\tilde{z}_y(x) - \tilde{z}_y^*(x))}{\exp(\tilde{z}_j(x) - \tilde{z}_j^*(x))} \leq \frac{1}{\pi_j} \frac{\exp(\hat{z}_y(x) - \hat{z}_y^*(x))}{\exp(\hat{z}_j(x) - \hat{z}_j^*(x))} + (1 + \frac{1}{\pi_j}) \exp(\epsilon) \tag{10}$$

must also satisfy Eq. (5) under $\pi_j = n_j/n, \forall j$. In other words, if the solution to the classical softmax loss is as diverse as

$$\hat{\mathcal{Z}} = \{\hat{z}(x) : \hat{z}(x) \text{ satisfies Eq. (5)}\}, \tag{11}$$

then the solution to the balanced softmax loss is at least as diverse as

$$\tilde{\mathcal{Z}} = \{\tilde{z}(x) : \text{there exists } \hat{z}(x) \in \hat{\mathcal{Z}} \text{ such that } \tilde{z}(x) \text{ satisfies Eq. (10)}\}, \tag{12}$$

We note that Eq. (10) is equivalent to

$$\frac{\tilde{q}_y(x)}{\tilde{q}_j(x)} / \frac{\tilde{q}_y^*(x)}{\tilde{q}_j^*(x)} \leq \frac{1}{\pi_j} \frac{\hat{q}_y(x)}{\hat{q}_j(x)} / \frac{\hat{q}_y^*(x)}{\hat{q}_j^*(x)} + (1 + \frac{1}{\pi_j}) \exp(\epsilon), \tag{13}$$

where $\hat{q}, \tilde{q}, q^*$ denotes the prediction probabilities (as opposed to logits) that correspond to $\hat{z}, \tilde{z}, z^*$. It implies that $\tilde{\mathcal{Z}}$ is at least as diverse as

$$\log \frac{\tilde{q}_j^*(x)}{\tilde{q}_y^*(x)} - \log \frac{\tilde{q}_j(x)}{\tilde{q}_y(x)} \leq \log \frac{\hat{q}_j^*(x)}{\hat{q}_y^*(x)} - \log \frac{\hat{q}_j(x)}{\hat{q}_y(x)} - C, \tag{14}$$

with $C = \log \pi_j$, since $(1 + \frac{1}{\pi_j}) \exp(\epsilon) > 0$. Thus, when $\pi_j < 1/K$ is small, then $\hat{q}$ in the balanced softmax loss is allowed to be much more diverse than in the classical softmax loss. It is because $\frac{\tilde{q}_y(x)}{\tilde{q}_j(x)} / \frac{\tilde{q}_y^*(x)}{\tilde{q}_j^*(x)}$, the deviation of $\hat{q}$ from the optimal solution q^* is allowed a much larger range than $\frac{\hat{q}_y(x)}{\hat{q}_j(x)} / \frac{\hat{q}_y^*(x)}{\hat{q}_j^*(x)}$.

The implications of diversity on ensemble NLLs. We next consider the implications of this diversity on the ensemble NLLs.

We first further rewrite the logit ensemble NLL and probability ensemble NLL:

$$\underbrace{-\mathop{\mathbb{E}}_{D}\left[\boldsymbol{y}^T \cdot \ln \bar{\boldsymbol{q}}\right]}_{\text{logit ensemble NLL}} = \underbrace{-\frac{1}{M}\sum_{i=1}^{M} \boldsymbol{y}^T \cdot \ln \boldsymbol{q}_i{}^*}_{\text{average bias}} + \underbrace{\frac{1}{M}\sum_{i=1}^{M} \mathop{\mathbb{E}}_{D}\left[D_{\mathrm{KL}}(\boldsymbol{q}_i^* || \boldsymbol{q}_i)\right]}_{\text{average variance}} + \underbrace{\mathop{\mathbb{E}}_{D}\left[\frac{1}{M}\sum_{j=1}^{K}\sum_{i=1}^{M} \bar{\boldsymbol{q}}^{(j)} [\log \frac{\boldsymbol{q}_i^{(j)}}{\bar{\boldsymbol{q}}^{(j)}}])\right]}_{\text{-diversity}} \tag{15}$$

$$\begin{aligned}\underbrace{-\mathop{\mathbb{E}}_{D}\left[\boldsymbol{y}^T \cdot \ln \boldsymbol{q}^\dagger\right]}_{\text{probability ensemble NLL}} &= -\frac{1}{M}\sum_{i=1}^{M} \boldsymbol{y}^T \cdot \ln \boldsymbol{q}_i{}^* + \frac{1}{M}\sum_{i=1}^{M} \mathop{\mathbb{E}}_{D}\left[D_{\mathrm{KL}}(\boldsymbol{q}_i^* || \boldsymbol{q}_i)\right] \\ &\quad + \underbrace{\mathop{\mathbb{E}}_{D}\left[\frac{1}{M}\sum_{i=1}^{M}\left[\log\left[\frac{q_i^{(y)}}{\frac{1}{M}\sum_{i=1}^{M} q_i^{(y)}}\right]\right]\right]}_{\text{-dependency}},\end{aligned} \tag{16}$$

Thus, the difference between logit ensemble NLL and probability ensemble NLL is

$$\text{Diff} = \mathop{\mathbb{E}}_{D}\left[\frac{1}{M}\sum_{j=1}^{K}\sum_{i=1}^{M} \bar{\boldsymbol{q}}^{(j)} [\log \frac{\boldsymbol{q}_i^{(j)}}{\bar{\boldsymbol{q}}^{(j)}}])\right] - \mathop{\mathbb{E}}_{D}\left[\frac{1}{M}\sum_{i=1}^{M}\left[\log\left[\frac{q_i^{(y)}}{\frac{1}{M}\sum_{i=1}^{M} q_i^{(y)}}\right]\right]\right] \tag{17}$$

$$= \mathop{\mathbb{E}}_{D}\left[\frac{1}{M}\sum_{j=1}^{K}\sum_{i=1}^{M} \bar{\boldsymbol{q}}^{(j)} [\log \frac{\boldsymbol{q}_i^{(j)}}{\boldsymbol{q}_i^{(y)}} - \log \frac{\bar{\boldsymbol{q}}^{(j)}}{\frac{1}{M}\sum_{i=1}^{M} q_i^{(y)}}])\right] \tag{18}$$

Proposition E.2 (NLL of ensemble models under balanced softmax loss)**.** *Suppose that the average probabilities $\bar{\boldsymbol{q}}^{(j)}$ and $\frac{1}{M}\sum_{i=1}^{M} q_i^{(y)}$ are close to the optimal prediction probabilities $\boldsymbol{q}^{*,(j)}$ and $q_i^{*,(y)}$ in the sense that $|\frac{\bar{\boldsymbol{q}}^{(j)}}{\frac{1}{M}\sum_{i=1}^{M} q_i^{(y)}} - \frac{\boldsymbol{q}^{*,(j)}}{q_i^{*,(y)}}| < \delta$. Further assume that the individual ensemble models can cover a β-substantial portion of the ϵ-approximate optimal solution set, namely Eq. (14) holds with $C = \beta \log \pi_j$. Then the NLL of the balanced softmax can outperform the NLL of the classical softmax,*

$$\textit{Diff}_{\textit{balanced}} - \textit{Diff}_{\textit{classical}} \leq \beta \sum_{j=1}^{K} \bar{\boldsymbol{q}}^{(j)} \log \pi_j - \log \delta.$$

Proposition E.2 is an immediate consequence of Eq. (14) and Eq. (18). It implies that the gain from logit ensembling is higher when π_j's take smaller values, especially when the data is long-tailed.