A Reproducibility Study of Decoupling Feature Extraction and Classification Layers for Calibrated Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Many neural networks, especially over-parameterized ones, suffer from poor calibration and overconfidence. To address this, Jordahn & Olmos (2024) recently proposed a Two-Stage Training (TST) procedure that decouples the training of feature extraction and classification layers. In this study, we replicate their findings and extend their work through a series of ablation studies. We reproduce their main results and find that most of them replicate, with slight deviation for CIFAR100. Additionally, we extend the author's results by exploring the impact of different model architectures, Monte Carlo (MC) sample sizes, and classification head designs. We further compare the method with focal loss - an implicit regularization technique known to improve calibration - and investigate whether calibration can be improved further by combining the two methods. We find that calibration can be improved even further by using focal loss in the first training stage of two-stage training. Our experiments validate the claims made by Jordahn & Olmos (2024), and show the transferability of the two-stage training to different architectures.

1 Introduction

Calibration is crucial for trustworthy machine learning, especially in safety-critical fields like medicine and self-driving cars, where understanding a model's uncertainty is key to assessing decision reliability. Neural networks, particularly over-parameterized ones, tend to be poorly calibrated and have an overconfidence bias. Efforts to improve calibration include post-hoc methods and modifications to training procedures (Wang, 2023).

In *Decoupling Feature Extraction and Classification Layers for Calibrated Neural Networks*, Jordahn & Olmos (2024) propose Two-Stage Training (TST) and Variational Two-Stage Training (V-TST), for calibration of Deep Neural Networks (DNNs). The methods decouple the training of the feature extraction and classification layers. At first, all layers are trained jointly from scratch and in the second stage of training, the last layer is re-initialized and re-trained from scratch while the weights of the feature extraction layers are kept frozen.

We conducted several experiments to analyze the performance of TST and V-TST under different configurations, and extend their method further by combining it with focal loss - both its constant and adaptive variant. Our main contributions are:

- Reproducing the main results (first two rows of Tables 1 and 2 of Jordahn & Olmos (2024)).
- Reproducing two of their ablation studies, namely applying their method to an under-parameterized CNN and a fine-tuned ViT (Tables 3 and 6 of Jordahn & Olmos (2024)).
- Conducting three ablation studies to investigate the dependence on model architecture, the sample size of a Monte Carlo (MC) estimation used in the second-stage of training and the architecture of the classification head.
- Further extending the original paper by comparing and combining the method with focal loss (Lin et al., 2017), another implicit regularization technique.

2 Related Work

The calibration of networks can be improved both post-hoc after training but also by changing the training procedure. Wang (2023) groups the different calibration techniques in four different categories: post-hoc calibration, regularization methods, uncertainty estimation, and composition methods. A widely used post-hoc calibration technique is Temperature Scaling (TS) (Guo et al., 2017) which is computationally efficient and simple and has been extended by other people (Wang, 2023).

In contrast to using post-hoc methods, there also exist many techniques focusing on training well-calibrated models, among which implicit regularization techniques are most relevant for this study. They include techniques such as data augmentation (Thulasidasan et al., 2020) and the use of different loss functions (Hebbalaguppe et al., 2022; Shamsi et al., 2023). An example of such an implicit technique is Focal Loss (Lin et al., 2017), defined as

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma}\log\,(p_t) \tag{1}$$

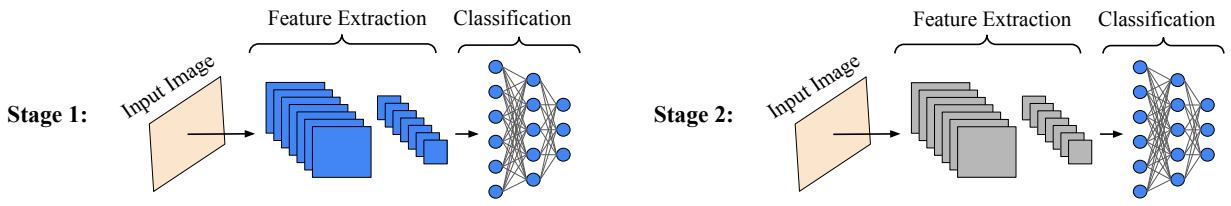
Focal loss is a modification of cross-entropy loss designed to address class imbalance. It introduces a weighting factor, $(1-p_t)^{\gamma}$, where p_t is the predicted probability of the true class. This factor down-weights the loss of well-classified examples ($p_t \approx 1$) and amplifies the influence of hard examples ($p_t \approx 0$), making training more effective. The focusing parameter $\gamma \geq 0$ determines the degree of this effect. Focal loss has been shown to be an upper bound on the regularized KL-divergence and improves calibration of networks independent of their architecture (see Mukhoti et al. (2020) for a more elaborate discussion on why focal loss improves calibration).

3 Methods

Jordahn & Olmos (2024) propose a novel two-stage training procedure for improved calibration of DNNs in classification tasks. Typically, the training of feature extraction layers and classification layers occurs jointly. Their findings indicate that training these layers separately in a two-stage fashion can improve model calibration. They argue that freezing the feature extraction layers limits the model's flexibility, preventing the classifier from artificially increasing label likelihood. In their paper, they present two variations of the two-stage training method: TST and V-TST.

3.1 TST

In TST, a model with initial parameters $\{\beta, \phi\}$, denoting the parameters of the feature extraction layers and classification layers respectively, is trained from scratch using cross-entropy loss (Jordahn & Olmos, 2024). The parameters of the feature extraction layers β are then frozen and the classification layers are re-initialized and re-trained from scratch using the same dataset D_{train} (see Fig.1). Further calibration improvements can be obtained by adjusting the dimensionality Z of the last hidden layer $\boldsymbol{z}$ of the classification layers, as the calibration performance varies with different choices of Z.



(a) First training stage: feature extraction and classification layers are trained jointly from scratch.

(b) Second training stage: feature extraction layers are frozen and classification layers are re-initialized and trained from scratch.

Figure 1: Illustration of training procedure for TST and V-TST. Blue indicates the layers are being trained while gray indicates frozen layers.

3.2 V-TST

V-TST poses additional regularization on $\boldsymbol{z}$. For the second stage, the parameters of the fully connected (FC) layers are optimized using the evidence lower-bound (ELBO):

$$\log p(\boldsymbol{y}) \geq \text{ELBO}_{\theta,\nu} = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})} \log[p(y|\boldsymbol{z})] - D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})). \tag{2}$$

They assume a standard normal Gaussian prior $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and an approximate posterior of the form $q(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\mu_{\beta,\theta}(\boldsymbol{x}), \sigma_{\beta,\theta}(\boldsymbol{x}))$. The predicted output label of the model is given by $p(y|\boldsymbol{z}) = \text{Cat}(\pi_\nu(\boldsymbol{z}))$. To approximate and sample from $q(\boldsymbol{z}|\boldsymbol{x})$, they apply the reparameterization trick $\boldsymbol{z} = \mu_{\beta,\theta}(\boldsymbol{x}) + \sigma_{\beta,\theta}(\boldsymbol{x}) \odot \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. This allows the expectation of $p(y|\boldsymbol{z})$ to be determined by MC sampling using m samples, as described in Kingma & Welling (2022).

3.3 Reproducing results

Using the code provided by the authors, we trained a WRN-28-10 architecture from scratch to reproduce the main results. We used the proposed methods TST and V-TST on the CIFAR10 and CIFAR100 datasets and included the evaluation on the shifted and out-of-distribution (OOD) data. We also reproduced two of the ablation studies in the original paper. The first ablation study evaluates the performance of two-stage training on an under-parameterized CNN architecture. Secondly, we performed the ablation study regarding two-stage training for pre-trained models using ViT-B/16 (Dosovitskiy et al., 2021) on Tiny ImageNet (Le & Yang, 2015).

3.4 Our extensions

In addition to reproducing the orginal results, we introduced a number of extensions to further investigate the performance of TST and V-TST. First, we examined the effectiveness of two-stage training with two additional CNN network architectures of comparable size to WRN-28-10 (36.5 million parameters) (Zagoruyko & Komodakis, 2017): ResNet-50 (He et al., 2015) and EfficientNet-B5 (Tan & Le, 2020) comprising of 25.6 million and 30.4 million parameters, respectively.

Next, we investigated different MLP configurations in the TST and V-TST setting, and examined the effect of varying the number of Monte Carlo samples used in the second stage of V-TST. While the original paper used a three-layer MLP for the classification layer, we experimented with replacing it with a four-layer MLP and a single-layer MLP, the latter being equivalent to re-initializing the final fully connected (FC) layer. Additionally, we explored whether increasing the number of MC samples in the second stage with V-TST from $m = 1$ (original proposed approach) to $m = 5$ could improve model calibration.

Lastly, as suggested in Jordahn & Olmos (2024), we investigated whether combining two-stage training with focal loss can lead to further enhancement in calibration performance.

4 Data

We used the same image datasets and preprocessing as the original paper, namely CIFAR10, CIFAR100 and Tiny ImageNet to train the models, and SVHN (OOD data) as well as CIFAR10-C and CIFAR100-C (shift data) to evaluate them. We randomly split the validation set of Tiny ImageNet into two halves to obtain validation and test sets, as the test set of the dataset is unlabeled. More details about the datasets and preprocessing steps can be found in Appendix A. For the ablation studies and the extensions, we used CIFAR10.

5 Experiments and Findings

Across all experiments, we evaluated the model performance using the metrics of the original paper, the expected calibration error (ECE) and the maximum calibration error (MCE) (Naeini et al., 2015), computed in percentage with 10 bins, on the regular and corrupted test sets. Additionally, we measured negative

log-likelihood (NLL) on both the training set and the test set, and included metrics AUROC for the shifted datasets and false-positive rate at 95% (FPR95) for the OOD datasets.

In all experiments, the models were trained using the Adam optimizer with learning rate 10^{-4} and cross-entropy (CE) loss to align with the original paper, except in the extensions where we used focal loss. In the first stage, the models were trained from scratch for 600 epochs with seed 1 unless otherwise specified. Throughout the paper, we refer to models after the first stage of training as *base models*. The TST and V-TST was done over 10 seeds (0-9) using 40 epochs and by re-initializing the final FC layer with a three-layer MLP head having dimensions [`output`, $3Z$], [$3Z$, Z] and [Z, #classes], where the `output` corresponds to the dimension of the model architecture's output before the classifier. A single NVIDIA T4 16GB GPU was used for all experiments.

5.1 Replication of Main Results

The replication of the main results is reported in Tables 1 and 2. For TST and V-TST, we used the same latent dimensions Z as the original paper. The CIFAR10 results closely align with those reported in Jordahn & Olmos (2024). The numbers are generally slightly worse for almost all metrics but lie within a close range to those found in the original paper. We believe that these slight differences can be attributed to the randomness involved in training the base model, leading to different base models to start from. While the authors specify running TST and V-TST over seeds 0-9, they do not specify the seed used for training the base model.

For CIFAR100, the results we obtained differ from those reported by Jordahn & Olmos (2024) in several aspects. First, we observed discrepancies in the baseline WRN models, which we trained for two different random seeds. Despite this, we were unable to replicate comparable results for either seed. Our baseline models demonstrated lower accuracy, but exhibited better calibration, as indicated by lower ECE and MCE values. For example, Jordahn & Olmos (2024) obtained a MCE of 82.02 for the baseline model, whereas our corresponding value was 27.52. Similar to the findings in Jordahn & Olmos (2024), we observed improvements in both calibration and accuracy when applying TST and V-TST, although the magnitude of these improvements differed. For instance, V-TST ($m = 10$) improved the ECE by 47.1%, whereas the original paper reported a 73.0% improvement. A similar trend was observed for TST. However, in contrast to Jordahn & Olmos (2024), our results showed greater improvements in accuracy and NLL when applying two-stage training. Finally, our experiments indicate that TST and V-TST do not outperform temperature scaling for CIFAR100.

We noticed that our base models for CIFAR100 achieved the lowest validation loss and were obtained in the early epochs, 18 and 16, even though the training accuracy reached around 100% and the validation accuracy kept improving to similar levels reported in the original paper. Similar results may have been obtainable by exploring more seeds or saving the model with the highest validation accuracy rather than the model with the lowest validation loss.

Consistent with Jordahn & Olmos (2024), we showed that V-TST and TST improve ECE and MCE across both datasets (see Tab. 1). Under distribution shifts, we similarly found that TST and V-TST significantly improve the calibration of the WRN and outperform temperature scaling for CIFAR10. However, unlike the original paper, our results indicate that TST generally performs slightly better than base models in OOD tasks. Additionally, we found that in the case of CIFAR100, TST and V-TST significantly improve the accuracy of a model that has overfitted and that temperature scaling achieves better calibration performance for the model.

5.1.1 Replication of CNN ablation study

To reproduce Table 3, we implemented a simple CNN, following the architecture detailed in the Appendix of the original paper. After training the base model, we applied TST and V-TST and matched the latent dimension with their results. Consistent with Jordahn & Olmos (2024), we found that two-stage training on an under-parameterized model worsens ECE but improves MCE, while TST can improve both as shown in Table 3. The lower proportion of frozen to trainable parameters in the second stage of an under-parameterized

Table 1: Replication of test metrics on in-distribution data test-sets. For V-TST, $m = 1$ and $m = 10$ indicate using 1 and 10 samples in the MC approximation, respectively. Temp. WRN is the temperature-scaled WRN. For CIFAR10, we used $Z = 128$ for V-TST ($m = 10$) and TST, and $Z = 32$ for V-TST ($m = 1$). For CIFAR100, $Z = 512$ was used for V-TST ($m = 10$) and TST, and $Z = 128$ for V-TST ($m = 1$). Bold indicates the best-performing model.

Dataset	Model	Accuracy	ECE	MCE	Train NLL	Test NLL
CIFAR10	V-TST, $m = 10$	92.44±0.02	**1.38±0.100**	**21.26±6.44**	0.0655±0.0004	0.2579±0.0017
	V-TST, $m = 1$	90.9±0.03	1.78±0.148	24.80±8.981	0.1116±0.0012	0.3732±0.0041
	TST	**92.55±0.01**	3.34±0.016	24.68±3.883	**0.0532±0.0002**	**0.2516±0.0007**
	Temp. WRN	92.52	4.70	25.2	0.0616	0.3024
	WRN	92.52	6.06	36.28	0.1031	0.5124
CIFAR100	V-TST, $m = 10$	**64.32±0.09**	9.92±0.288	21.74±1.349	**0.5008±0.0082**	**1.4159±0.0034**
	V-TST, $m = 1$	62.35 ± 0.07	10.71 ± 0.318	20.48 ± 0.780	0.6064 ± 0.0049	1.5634 ± 0.0045
	TST	61.82±0.16	10.32±0.560	20.55±2.404	0.6448±0.0098	1.5212±0.0067
	Temp. WRN	52.31	**2.72**	**6.42**	1.2029	1.8358
	WRN	52.31	17.38	27.52	1.1451	2.0083
	WRN - 2nd seed	51.54	14.72	27.03	1.2923	2.0009

Table 2: Replication of evaluation metrics on shifted data and OOD data.

Dataset	Model	SHIFT ECE	SHIFT MCE	OOD AUROC	OOD FPR95
CIFAR10	V-TST, $m = 10$	**9.41±0.3**	20.33±0.66	0.851±0.003	0.702±0.002
	V-TST, $m = 1$	12.53±0.29	**20.19±0.56**	0.766±0.003	0.778±0.003
	TST	13.09±0.04	29.32±0.10	0.883±0.001	**0.688±0.003**
	Temp. WRN	16.44	37.93	**0.885**	0.698
	WRN	20.06	45.62	0.874	0.708
CIFAR100	V-TST, $m = 10$	20.30 ± 0.42	32.39 ± 0.66	**0.727 ± 0.002**	**0.817 ± 0.002**
	V-TST, $m = 1$	21.15 ± 0.40	34.01 ± 0.60	0.717 ± 0.002	0.828 ± 0.002
	TST	21.13 ± 0.76	30.82 ± 1.13	0.692 ± 0.002	0.846 ± 0.004
	Temp. WRN	**11.09**	**20.46**	0.677	0.878
	WRN	28.15	43.18	0.658	0.885
	WRN - 2nd seed	26.10	39.61	0.655	0.858

model increases flexibility, possibly explaining the ineffectiveness of two-stage training. Additionally, we observed that for TST, the authors report improved ECE with $Z = 32$, the only latent dimension where ECE improved. With $Z = 128$, we found the ECE to be worse than the base model.

5.1.2 Replication of ViT ablation study

The other ablation study concerned applying the two-stage training after fine-tuning a pre-trained ViT. Implementation details are described in Appendix B. Again, our results were generally consistent with their findings. It can be seen from Table 3 that using V-TST can improve the ECE and MCE while TST worsens both ECE and MCE. They stress the good regularization properties of V-TST and also conclude that training feature extraction layers and classification layers together from scratch is important for the two-stage training to be effective.

5.2 Ablation Study: Dependence on Network Architecture

We adapted the architectures of ResNet-50 and EfficientNet-B5 to fit the smaller image size of 32×32 of CIFAR10, for details we refer to Appendix B. The architectures were trained from scratch and then we applied TST and V-TST.

Both TST and V-TST improved calibration for ResNet-50 and EfficientNet-B5 (see Tab.5). The best-performing model for ResNet-50 was obtained with V-TST ($m = 10$), which aligns with the results for WRN-28-10. V-TST ($m = 10$) improved ECE and MCE by 88.14% and 62.17%, respectively. This model also

Table 3: Replication of test metrics for ablation studies on in-distribution data test-sets using a simple CNN on CIFAR10. We used $Z = 512$ for V-TST with $m = 10$, $Z = 128$ for V-TST with $m = 1$, and $Z = 32$ for TST.

Model	**Accuracy**	**ECE**	**MCE**
V-TST, $m = 10$	**71.25±0.04**	1.82±0.07	8.85±1.42
V-TST, $m = 1$	67.85±0.07	1.25±0.08	**6.55±1.17**
TST, $Z = 128$	**71.03±0.04**	1.11±0.05	18.64±0.49
TST	70.55±0.05	**0.73±0.05**	14.80±1.98
CNN	70.83	0.80	19.46

Table 4: Replication of test metrics for ablation studies on in-distribution data test-sets while fine-tuning a ViT on Tiny ImageNet. We used $Z = 128$ for V-TST and TST.

Model	**Accuracy**	**ECE**	**MCE**
V-TST, $m = 10$	**84.36±0.04**	3.35±0.07	12.46±0.81
V-TST, $m = 1$	83.64±0.08	**1.56±0.08**	**11.13±1.17**
TST	83.98±0.05	6.11±0.11	22.03±2.03
FT ViT-B/16	83.38	3.33	13.60

reached the highest accuracy of 89.80%. For EfficientNet-B5, the best calibration is obtained for temperature scaling, which improved ECE and MCE by 90.74% and 92.50%, respectively. The best-performing two-stage procedure for EfficientNet-B5 was V-TST ($m = 1$), for which ECE and MCE was improved by 40.93% and 81.42%.

Calibration was also improved when using TST and V-TST on shifted data. For ResNet-50, the best performance is obtained for V-TST ($m = 10$), while temperature scaling outperforms two-stage training for EfficientNet-B5. Similar to the paper, a decrease in performance occurred for the OOD data for ResNet-50 when using two-stage training. The contrary was observed for EfficientNet-B5.

The similarity in the results between WRN and ResNet may be due to their architectural resemblance, as WRN employs standard residual blocks with increased width and reduced depth compared to ResNet (Zagoruyko & Komodakis, 2017). In contrast, EfficientNet differs more from WRN, by using features like compound scaling (Tan & Le, 2020). These architectural differences may explain why the results achieved with WRN did not transfer in the same way to EfficientNet. For further improvements in the results for EfficientNet when using TST and V-TST, different latent dimensions could be explored.

Table 5: Dependence on TST and V-TST on network architecture. The latent dimensions used for TST, V-TST ($m = 1$), and V-TST ($m = 10$) were the same as those used for WRN-28-10, namely $Z = 128$, $Z = 32$ and $Z = 128$, respectively.

Model	**Accuracy**	**ECE**	**MCE**	**SHIFT ECE**	**SHIFT MCE**	**OOD AUROC**	**OOD FPR95**
V-TST, $m = 10$	**89.80 ± 0.03**	**0.95 ± 0.081**	**13.15 ± 2.07**	**6.36 ± 0.24**	**11.17 ± 0.38**	0.776 ± 0.004	0.810 ± 0.003
V-TST, $m = 1$	86.77 ± 0.09	1.60 ± 0.075	30.52 ± 6.829	10.95 ± 0.40	15.40 ± 0.63	0.719 ± 0.001	0.852 ± 0.002
TST	89.77 ± 0.02	4.19 ± 0.016	18.25 ± 1.670	13.38 ± 0.06	25.94 ± 0.10	0.845 ± 0.001	0.753 ± 0.004
Temp. ResNet-50	89.74	5.82	22.09	16.83	33.82	**0.861**	**0.744**
ResNet-50	89.74	8.01	34.75	21.58	44.94	0.852	0.762
V-TST, $m = 10$	76.93 ± 0.04	4.97 ± 0.267	16.40 ± 1.926	13.90 ± 0.34	20.73 ± 0.55	0.683 ± 0.001	**0.905 ± 0.001**
V-TST, $m = 1$	74.87 ± 0.08	4.85 ± 0.270	14.99 ± 1.615	14.54 ± 0.26	19.91 ± 0.36	0.663 ± 0.001	0.906 ± 0.002
TST	**77.10 ± 0.04**	7.20 ± 0.225	43.68 ± 8.575	16.54 ± 0.30	25.43 ± 0.47	**0.689 ± 0.001**	0.909 ± 0.001
Temp. EfficientNet-B5	76.48	**0.76**	**6.05**	**8.95**	**12.80**	0.681	0.920
EfficientNet-B5	76.48	8.21	80.68	18.58	28.35	0.678	0.917

5.3 Ablation Study: Second Stage Network Architecture

To examine the impact of MLP head size used in the second stage of training for TST and V-TST, we conducted experiments by varying the original three-layer MLP head to a single-layer and a four-layer MLP, while keeping the latent dimensions consistent with the replication experiments in section 5.1 across all setups. For further details, we refer to Appendix B.

The results in Table 6 and Table 7 show that TST and V-TST with a four-layer MLP improved ECE and MCE with and without distribution shifts. Our obtained results are close to the ones for a three layer MLP but slightly worse. For TST, the lower train NLL and higher test NLL show signs of more overfitting in the four-layer configuration. We believe that the additional layer without optimizing the dimension of Z introduces too many trainable parameters. We also noticed that the models converged faster with four layers. Furthermore, the results of TST using a four-layered MLP but with a decreased latent dimension of $Z = 10$ show comparable results to the ones with a three layer MLP and latent dimension of $Z = 128$, highlighting the potential importance of balancing number of layers in the MLP and the latent dimension.

In the single-layer configuration, the TST results show that re-training the final FC layer improves the MCE by 49.37% and all the shift and OOD metrics, still outperforming temperature scaling in most metrics. This shows that simply re-training the FC layer can effectively enhance model calibration and robustness under distribution shifts. However, the single-layer MLP performs worse than the multi-layered ones in terms of ECE and MCE scores for in distribution data, likely due to the reduced capacity to capture complex calibration patterns.

Table 6: Test metrics on in-distribution data for the effect of the size of the MLP in the second stage of training. The value l indicates the number of layers in the MLP head of TST and V-TST. We used $Z = 128$ as latent dimension unless stated otherwise.

Model	Accuracy	ECE	MCE	Train NLL	Test NLL
V-TST, $l = 4$, $m = 10$	92.50±0.02	1.95±0.141	26.08±5.912	0.0733±0.0006	0.2823±0.0029
V-TST, $l = 4$, $m = 1$	91.65±0.05	**1.71±0.029**	27.94±8.612	0.1227±0.0011	0.3971±0.0026
TST, $l = 4$	**92.52±0.02**	3.50±0.280	23.36±2.717	0.0547±0.0002	0.2588±0.0010
TST, $l = 4$, $Z = 10$	91.64±0.11	2.45±0.150	**14.74±0.997**	0.0702±0.0021	0.3052±0.0091
TST, $l = 1$	**92.52±0.02**	6.69±1.203	18.37±2.763	**0.0526±0.00001**	**0.2408±0.0016**

Table 7: Shift and OOD test metrics for the effect of the size of the MLP in the second stage of training. The value l indicates the number of layers in the MLP head of TST and V-TST. We used $Z = 128$ as latent dimension unless stated otherwise.

Model	SHIFT ECE	SHIFT MCE	OOD AUROC	OOD FPR95
V-TST, $l = 4$, $m = 10$	11.86±0.39	28.16±0.99	0.802±0.007	0.695±0.003
V-TST, $l = 4$, $m = 1$	12.47±0.29	**17.07±0.47**	0.71±0.004	0.773±0.004
TST, $l = 4$	13.45±0.04	30.25±0.09	0.875±0.002	0.727±0.013
TST, $l = 4$, $Z = 10$	11.69±0.43	24.54±0.90	0.883±0.001	0.688±0.003
TST, $l = 1$	**11.46±0.27**	25.35±0.65	**0.884±0.001**	**0.682±0.002**

5.4 Ablation Study: V-TST Dependence on Number of Samples in Training

We also investigated whether increasing the number of samples during V-TST affects the model performance. Specifically, we re-trained the base WRN for CIFAR10 using V-TST with $n = 5$ samples and compared it to the original setup with $n = 1$. As shown in Table 8, using $n = 5$ results in slightly better MCE and OOD metrics, while overall calibration performance remains comparable or slightly worse. In contrast to our expectations, increasing the number of samples drawn during training did not prove to be beneficial for improving the calibration. However, we suspect that further exploration with different values would provide

deeper insights to draw better conclusions. We also noticed that the variability in most of the evaluation metrics reduces with $n = 5$.

Table 8: Evaluation metrics on in-distribution, shifted data, and OOD data for the effect of the number of samples used in the second stage of training with V-TST. We used $Z = 128$ as latent dimension.

Model	Accuracy	ECE	MCE	SHIFT ECE	SHIFT MCE	OOD AUROC	OOD FPR95
V-TST, $n = 5$, $m = 10$	**92.45±0.02**	**1.53±0.081**	**13.90±1.979**	**9.65±0.21**	20.55±0.48	**0.852±0.002**	**0.697±0.002**
V-TST, $n = 5$, $m = 1$	91.04±0.06	1.62±0.103	18.89±6.873	12.57±0.22	**20.08±0.39**	0.762±0.004	0.779±0.004

5.5 Extending the paper: Focal Loss

We compared the performance of two-stage training with focal loss, another implicit calibration method, and investigated if the performance of two-stage training can be improved by combining it with focal loss in several ways. We experimented with three configurations: using focal loss in both training stages, the first training stage only or the second training stage only. We also experimented with two different versions of focal loss: using a constant value for γ (FL) or an adaptive, sample-dependent schedule (FLA) as described in Mukhoti et al. (2020). In accordance with Mukhoti et al. (2020), we used $\gamma = 3$ for the constant loss and a sample-dependent scheme that uses $\gamma = 3$ for $\hat{p}_y \in [0.2, 1]$ and $\gamma = 5$ $\hat{p}_y \in [0, 0.2)$, where $\hat{p}_y$ denotes the probability of the network to predict the correct label y.

Focal loss base models First, we observed that a base model trained on constant focal loss (FL-B) or adaptive focal loss (FLA-B) is more calibrated on in-distribution and shifted data than a base model trained using cross-entropy loss (CE-B), demonstrating the expected calibration benefits of focal loss. However, V-TST with $m = 10$ and CE loss (V10-CE2) outperforms base models trained using focal loss (FL-B and FLA-B) in terms of both ECE and MCE scores as well as accuracy.

Focal loss in two-stage training When combining the two methods we noticed several key observations. Firstly, using focal loss as training objective for both stages in two-stage training significantly worsens the ECE score compared to the base models trained with focal loss (FL-B, FLA-B). The MCE score is also worsened when using adaptive focal loss compared to FLA-B, while it generally improves for constant focal loss. Comparing two-stage training performed using CE loss to focal loss in both stages, we noticed a similar trend. Two-stage training models using focal loss have higher ECE scores compared to corresponding models trained using CE loss (see Tab. 9). No clear trend can be seen for the MCE score. Hence, fully combining two-stage training with focal loss does generally not lead to improved calibration compared to a base model trained with focal loss or two stage training using CE loss.

Focal loss base and cross-entropy in second-stage Using focal loss in the second stage on top of a focal loss base model worsened the calibration overall. Therefore, we tested whether combining a better-calibrated base model (compared to a base model trained with CE loss, see Tab. 9) with a second stage that uses CE loss would enhance calibration performance. We found that this combination (V10-FLA-CE, V1-FLA-CE, FLA-CE) consistently improved both ECE and MCE significantly compared to the focal loss base models (FL-B, FLA-B) and the corresponding CE baseline models (V10-CE2, V1-CE2, CE2) while maintaining a similar accuracy. For instance, including FLA in the first stage for TST (FLA-CE) improved ECE and MCE by 37.1% and 27.4%, respectively (see Tab. 9). Results for using FL-B as a base model and performing second stage training with CE were similar to those using FLA-B as base model and are thus reported in the Appendix C.

Cross-entropy loss base and focal loss in second-stage For completeness, we also tested how using constant focal loss in the second stage on top of a CE base model would perform (V10-CE-FL, V1-CE-FL, CE-FL) and found a similar behavior to using constant focal loss or adaptive focal loss in both stages of training (V10-FL2, V1-FL2, FL2, V10-FLA2, V1-FLA2, FLA2). Both ECE and MCE scores were higher than those obtained for CE-only two-stage training. Therefore, using focal loss in the second stage (regardless of whether focal loss or CE was used in the first stage) generally performs badly on in-distribution data.

Focal loss performance on shifted data Incorporating focal loss in two-stage training in any form leads to significantly lower ECE and MCE for shifted data compared to focal loss base models and two-stage CE models (see Tab. 10). The only exceptions to this are the V-TST models that used adaptive focal loss in the first stage and CE loss in the second stage (V10-FLA-CE, V1-FLA-CE) but even these perform similarly well compared to V-TST models that only use CE loss (V10-CE2, V1-CE2). Another interesting observation was that models trained in a two-stage fashion with focal loss in both or one of the stages perform better on shifted data than in-distribution data. The exception for this trend were models trained with adaptive focal loss in the first stage and CE loss in the second stage (V10-FLA-CE, V1-FLA-CE, FLA-CE). This is a rather surprising finding as the WRN-28-10 trained with CE loss (V10-CE2, V1-CE2, CE2) was better calibrated for in-distribution than shifted data for both CIFAR10 and CIFAR100 (see Tab. 1).

A possible explanation for the better performance on shifted data is that focal loss emphasizes hard examples during training (Lin et al., 2017), leading to better calibration on challenging shifted data but poorer calibration on easier in-distribution data. Additionally, focal loss with a fixed γ has different dynamics than the CE loss. Initially, models trained with focal loss show higher weight norms than those trained with CE loss. However, this trend reverse after furhter training so that the weight norms of the focal loss model become lower, which can be explained by the regularizing effect the focal loss has on the network weights (Mukhoti et al., 2020).

Focal loss performance on OOD data For the OOD data, we find two interesting results. First, TST generally performs best both in terms of AUROC and FPR95, regardless of the loss function combination (even the CE baseline model (CE-B) has the lowest FPR95 score for the TST). This shows the superiority of TST on OOD data. In contrast, V-TST ($m = 1$) performed worst in all configurations for both AUROC and FPR95. Second, among the TST models, using adaptive focal loss in both stages of training (FLA2) reaches the highest AUROC and lowest FPR95 rate overall, outperforming both the CE baseline and the other focal loss combinations. Thirdly, the combination of adaptive focal loss in the first stage and CE loss in the second stage (V10-FLA-CE, V1-FLA-CE, FLA-CE) mostly worsens the performance (both AUROC and FPR95) compared to using the adaptive focal loss in the first and second stages of training (V10-FLA2, V1-FLA2, FLA2) or the CE baseline (CE-B).

Table 9: Test metrics and evaluation metrics on in-distribution data for the effect of including Focal Loss (FL) and Adaptive Focal Loss (FLA) in the training procedure. The first loss before + sign indicates the loss used for training the base model prior to TST and V-TST. We used a latent dimension of $Z = 128$.

Model	Loss	Model Name	Accuracy	ECE	MCE	Train NLL	Test NLL
V-TST, $m = 10$	3 FL	V10-FL2	**91.77±0.03**	11.72±0.148	20.94±0.934	0.2016±0.0018	0.3694±0.0016
V-TST, $m = 1$	3 FL	V1-FL2	87.71±0.11	11.08±0.482	24.35±1.850	0.3128±0.0069	0.5185±0.0054
TST	3 FL	FL2	91.75±0.03	5.90±0.384	29.24±8.359	0.1261±0.0043	0.2796±0.0034
Temp. WRN	3 FL	FL-TS	**91.78**	1.89	**10.25**	0.0880	**0.2524**
WRN	3 FL	FL-B	**91.78**	**1.57**	27.45	**0.0688**	0.2604
V-TST, $m = 10$	5-3 FLA	V10-FLA2	92.02±0.03	11.82±0.151	23.31±1.625	0.1964±0.0016	0.3692±0.0012
V-TST, $m = 1$	5-3 FLA	V1-FLA2	87.73±0.08	12.08±0.327	23.75±0.839	0.3181±0.0044	0.5259±0.0031
TST	5-3 FLA	FLA2	92.06±0.04	5.96±0.340	27.73±6.487	0.1199±0.0035	0.2735±0.0030
Temp. WRN	5-3 FLA	FLA-TS	**92.14**	**1.42**	81.25	0.077	**0.242**
WRN	5-3 FLA	FLA-B	**92.14**	1.97	**22.83**	**0.0602**	0.2539
V-TST, $m = 10$	CE + 3 FL	V10-CE-FL	92.40±0.02	9.84±0.235	26.92±6.064	0.1659±0.0022	0.3316±0.0020
V-TST, $m = 1$	CE + 3 FL	V1-CE-FL	89.26±0.08	10.65±0.403	**25.35±2.224**	0.2667±0.0037	0.4703±0.0034
TST	CE + 3 FL	CE-FL	**92.54 ± 0.02**	**3.21 ± 0.255**	27.50 ± 4.491	**0.0822 ± 0.0021**	**0.2483 ± 0.0024**
V-TST, $m = 10$	5-3 FLA + CE	V10-FLA-CE	**92.04±0.01**	**0.80±0.045**	20.33±1.981	0.0775±0.0004	0.2861±0.0008
V-TST, $m = 1$	5-3 FLA + CE	V1-FLA-CE	90.11 ± 0.04	1.52 ± 0.129	20.42 ± 6.653	0.1351 ± 0.0012	0.4101 ± 0.0038
TST	5-3 FLA + CE	FLA-CE	92.13±0.03	2.10±0.032	**14.92±2.292**	**0.0595±0.0001**	**0.2529±0.0004**
V-TST, $m = 10$	CE	V10-CE2	92.44±0.02	**1.38±0.100**	**21.26±6.44**	0.0655±0.0004	0.2579±0.0017
V-TST, $m = 1$	CE	V1-CE2	90.9±0.03	1.78±0.148	24.80±8.981	0.1116±0.0012	0.3732±0.0041
TST	CE	CE2	**92.55±0.01**	3.34±0.016	24.68±3.883	**0.0532±0.0002**	**0.2516±0.0007**
Temp. WRN	CE	CE-TS	92.52	4.7	25.2	0.0616	0.3024
WRN	CE	CE-B	92.52	6.06	36.28	0.1031	0.5124

Table 10: Evaluation metrics on shifted and OOD data for the effect of including Focal Loss (FL) and Adaptive Focal Loss (FLA) in the training procedure. The first loss before + sign indicates the loss used for training the base model prior to TST and V-TST. We used a latent dimension of $Z = 128$.

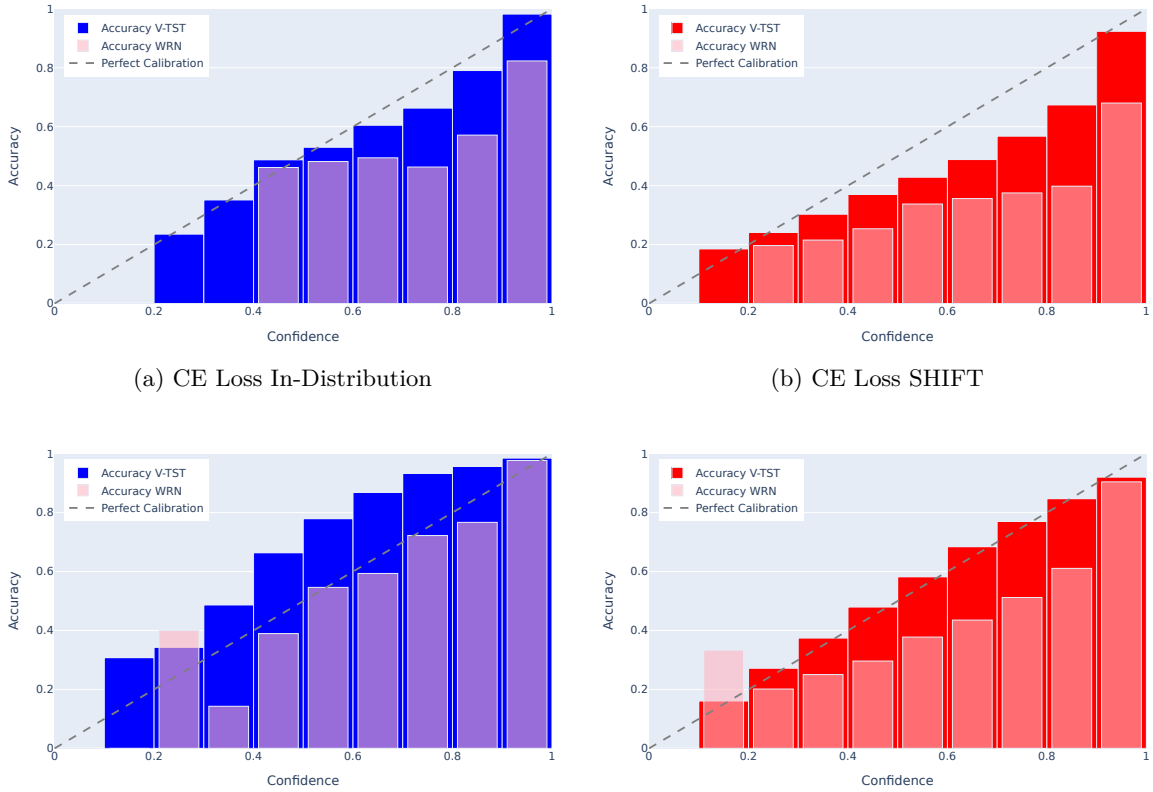
Model	Loss	Name	Shift ECE	Shift MCE	OOD AUROC	OOD FPR95
V-TST, $m = 10$	3 FL	V10-FL2	3.92±0.39	6.07±0.42	0.799±0.004	0.768±0.003
V-TST, $m = 1$	3 FL	V1-FL2	2.96±0.15	4.97±0.31	0.701±0.002	0.850±0.003
TST	3 FL	FL2	**2.44±0.10**	**4.75±0.29**	**0.876±0.001**	**0.672±0.004**
Temp. WRN	3 FL	FL-TS	6.28	11.89	**0.875**	0.701
WRN	3 FL	FL-B	11.93	22.21	0.866	0.740
V-TST, $m = 10$	5-3 FLA	V10-FLA2	4.13±0.36	6.79±0.42	0.828±0.004	0.714±0.004
V-TST, $m = 1$	5-3 FLA	V1-FLA2	**2.50±0.15**	**4.28±0.29**	0.735±0.003	0.827±0.005
TST	5-3 FLA	FLA2	2.75±0.06	5.53±0.22	**0.887±0.001**	**0.625±0.004**
Temp. WRN	5-3 FLA	FLA-TS	7.06	13.36	0.88	0.65
WRN	5-3 FLA	FLA-B	12.74	24.30	0.88	0.69
V-TST, $m = 10$	CE + 3 FL	V10-CE-FL	3.92±0.39	6.07±0.42	0.799±0.004	0.768±0.003
V-TST, $m = 1$	CE + 3 FL	V1-CE-FL	**2.25±0.27**	**4.20±0.51**	0.745±0.004	0.810±0.005
TST	CE + 3 FL	CE-FL	2.44±0.10	4.75±0.29	**0.876±0.001**	**0.672±0.004**
V-TST, $m = 10$	5-3 FLA + CE	V10-FLA-CE	**10.01±0.16**	19.66±0.37	0.825±0.002	0.727±0.003
V-TST, $m = 1$	5-3 FLA + CE	V1-FLA-CE	12.65 ± 0.32	**19.19 ± 0.56**	0.762±0.002	0.790±0.003
TST	5-3 FLA + CE	FLA-CE	11.75±0.07	23.36±0.16	**0.877±0.001**	**0.676±0.002**
V-TST, $m = 10$	CE	V10-CE2	**9.41±0.3**	20.33±0.66	0.851±0.003	0.702±0.002
V-TST, $m = 1$	CE	V1-CE2	12.53±0.29	**20.19±0.56**	0.766±0.003	0.778±0.003
TST	CE	CE2	13.09±0.04	29.32±0.10	0.883±0.001	**0.688±0.003**
Temp. WRN	CE	CE-TS	16.44	37.93	**0.885**	0.698
WRN	CE	CE-B	20.06	45.62	0.874	0.708

Reliability diagrams Figure 2 shows the ECE plots for the base and V-TST models trained with CE loss and the 5-3 focal loss in both stages. Starting from an uncalibrated WRN model trained with CE loss in 2a (CE-B), V-TST in the second stage leads to an improved calibration (V10-CE2). 2c shows that the base WRN that was trained with adaptive focal loss (FLA-B) is already better calibrated than the base WRN model trained with CE (CE-B) in 2a. The second training stage with adaptive focal loss makes the model underconfident in its predictions but also evens out the overconfidence it showed for images that were around 35% certain. For adaptive focal loss on the shifted dataset (2d), both the base WRN (FLA-B) and V-TST (V1-FLA-B) models perform worse than on in-distribution data, with a decrease in accuracy. However, for the V-TST model (V1-FLA-B), this results in its confidence aligning more closely with its accuracy, whereas the base model becomes overconfident.

6 Conclusion

In this paper, we aimed to reproduce the main results from Jordahn & Olmos (2024). We confirmed their findings that decoupling the training of feature and classification layers improves the calibration of a WRN-28-10 model on CIFAR10 and CIFAR100, with slightly varying effects on CIFAR100. Additionally, we successfully reproduced two of their ablation studies — the CNN ablation study and the ViT fine-tuning ablation study. To further explore two-stage training, we conducted additional ablation studies, examining its performance on different model architectures, the impact of the second-stage network architecture, and the effect of varying the number of MC samples used during second-stage training. We found that the effects of improved calibration of two-stage training transfer to other similar model architectures, namely ResNet50 and EfficientNet-B5. For the second stage architecture, we observed the importance of balancing the number of layers in the MLP with the latent dimension Z used. We found that using more MC samples during training did not affect the calibration performance of the models positively.

Finally, we followed the suggestion of Jordahn & Olmos (2024) to combine their method with other calibration methods and investigated whether using it together with focal loss could further improve calibration. Overall, we found that the focal loss base models (FL-B, FLA-B) are better calibrated than the CE base models (CE-



(a) CE Loss In-Distribution

(b) CE Loss SHIFT

(c) Focal Loss In-Distribution

(d) Focal Loss SHIFT

Figure 2: ECE plots for models with CE and focal loss. (a) In-distribution ECE plot of a V-TST trained model ($Z = 128$, $m = 10$, V10-CE2) and the base WRN trained on CIFAR10 with CE loss (CE-B). (b) Same models with CE loss evaluated on shifted data CIFAR10-C. (c) In-distribution ECE plot of a V-TST trained model ($Z = 32$, $m = 1$, V1-FLA-B) and the base WRN trained on CIFAR10 with focal loss (FLA-B). (d) Same models with focal loss evaluated on shifted data CIFAR10-C.

B), as expected. However, using a second stage training with focal loss on top of the focal loss base models (V10-FL2, V1-FL2, FL2 or V10-FLA2, V1-FLA2, FLA2) performs worse than the focal loss base models alone (FL-B, FLA-B) and the baseline CE two-stage training models (V10-CE2, V1-CE2, CE2). In contrast, using CE loss on top of the focal loss base model (V10-FLA-CE, V1-FLA-CE, FLA-CE) in the second training stage outperforms both the focal loss base models (FL-B, FLA-B) and the baseline CE two-stage training model (V10-CE2, V1-CE2, CE2, CE-TS, CE-B). This was true regardless of whether adaptive or constant focal loss was used. Interestingly, for shifted data, this pattern is partly reversed. Including focal loss in either the first, the second or both stages improved both ECE and MCE compared to the CE baseline model (CE-B). Models using focal loss were almost always better calibrated than the CE two-stage training models (V10-CE2, V1-CE2, CE2). Especially the models that used focal loss in the second stage (regardless of whether they also used it in the first stage) performed significantly better on shifted data than the corresponding two-stage training CE model. The combination that had performed best on in-distribution data, namely using a CE loss on top of the focal loss base model (V10-FLA-CE, V1-FLA-CE, FLA-CE) performed worst on shifted data compared to the other models that integrate focal loss, reaching similar or higher ECE and MCE values than the corresponding CE baseline models. Thus, we find the interesting pattern that using focal loss in the first stage of training is beneficial for in-distribution data whereas for the shifted data using focal loss in the second or both stages outperforms the CE baseline. Given our obtained results, the best results on in-distribution data can be achieved when training a model with focal loss in the first stage and using CE in the second stage in a V-TST fashion (V10-FLA-CE, V1-FLA-CE). For OOD

data we noticed that for all models TST reaches the highest calibration while V-TST ($m = 1$) results in the lowest calibration.

7 Future Work

Future extensions should investigate possible explanations as to why using focal loss for both stages of training decreased model calibration compared to the focal loss base model. One possible next step could be to apply early stopping based on ECE (as has been tested by Mukhoti et al. (2020)) during the second stage of training to avoid an over-regularization of the weights. It should also be tested whether the focal loss results we obtained are transferable to other model architectures apart from WRN-28-10. Another idea would be to experiment further with the latent dimension like trying higher latent dimensions for the MLP in the second stage as we only used either $Z = 128$ or $Z = 32$. It would also be interesting to see if a similar result pattern would be obtained for other datasets with more classes like CIFAR100.

Additionally, we would be interested to see further work investigating how the TST and V-TST perform when combined with other implicit regularization techniques such as OKO (Muttenthaler et al., 2023) which had also been suggested in Jordahn & Olmos (2024). They claimed that their method is not an alternative to other techniques but rather complements them, however, our results show that combining TST and V-TST with other calibration techniques is not always straightforward but requires careful experimental design. Similarly, it could be explored how calibration is affected by using either more or less data augmentation than has been used by Jordahn & Olmos (2024) and us.

8 Challenges and Limitations

During the replication study we encountered some challenges in replicating the results reported in Jordahn & Olmos (2024) due to missing information and incomplete code, which required us to re-implement parts of it and make assumptions about the training hyperparameters. For instance, details on the MLP design for the CNN in TST and V-TST, choices for latent dimension, the seed used for training the base WRNs, and information regarding the pre-trained model for ViT fine-tuning were not provided. Additionally, the model structures reported in Tables 1 and 2 were not immediately apparent. Variations in the latent dimension Z for each model could only be seen by inspecting and comparing all results in the Appendix in detail.

The theoretical justification for the proposed methods by Jordahn & Olmos (2024) could have been discussed in more detail. While the authors provide some intuition regarding why the decoupled learning of feature and classification layers improves calibration, they offer limited reasoning for selecting the latent dimension. Although they conduct multiple experiments with different Z, the reasoning behind the choice of latent dimension is not fully explained, even though different values of Z lead to significant variation in the results.

References

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL `https://arxiv.org/abs/1512.03385`.

Ramya Hebbalaguppe, Rishabh Patra, Tirtharaj Dash, Gautam Shroff, and Lovekesh Vig. Calibrating deep neural networks using explicit regularisation and dynamic data pruning, 2022. URL `https://arxiv.org/abs/2212.10005`.

Mikkel Jordahn and Pablo M. Olmos. Decoupling feature extraction and classification layers for calibrated neural networks, 2024. URL https://arxiv.org/abs/2405.01196.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge, 2015. URL https://api.semanticscholar.org/CorpusID:16664790.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.

Lukas Muttenthaler, Robert A Vandermeulen, Qiuyi Zhang, Thomas Unterthiner, and Klaus-Robert Müller. Set learning for accurate and calibrated models. *arXiv preprint arXiv:2307.02245*, 2023.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2901–2907. AAAI Press, 2015. ISBN 0262511290.

Afshar Shamsi, Hamzeh Asgharnezhad, AmirReza Tajally, Saeid Nahavandi, and Henry Leung. An uncertainty-aware loss function for training neural networks with calibrated predictions, 2023. URL https://arxiv.org/abs/2110.03260.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. URL https://arxiv.org/abs/1905.11946.

Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks, 2020. URL https://arxiv.org/abs/1905.11001.

Cheng Wang. Calibration in deep learning: A survey of the state-of-the-art. *arXiv preprint arXiv:2308.01222*, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017. URL https://arxiv.org/abs/1605.07146.

A Datasets

Table 11: Overview of used datasets and corresponding preprocessing steps.

Dataset	# Images	# Classes	Size	Preprocessing	Implementation
CIFAR10	60,000	10	32x32	Normalization, Random Crop, Random Horizontal Flip	PyTorch
CIFAR100	60,000	100	32x32	Normalization, Random Crop, Random Horizontal Flip	PyTorch
SVHN	600,000	10	32x32	Normalization	PyTorch
CIFAR10-C	60,000	10	32x32	Corruption Added	Downloaded
CIFAR100-C	60,000	100	32x32	Corruption Added	Downloaded
Tiny ImageNet	100,000	200	64x64	Resize (224x224), Normalization	GitHub Repo

B Network architeture details

ViT architecture As the original study does not specify which weights or latent dimension they used, we decided to use the weights of google/vit-base-patch16-224-in21k from the transformers package which corresponds to the ViT-B/16 architecture and used the latent dimension $Z = 128$ for both V-TST and TST. As Tiny ImageNet images are too small for the ViT, we had to resize them before using them as an input to the model. We used the ViTForImageClassification and the ViTModel class from the transformer package to include the ViT in the code base of the original paper.

ResNet-50 and EfficientNet-B5 architecture The architectures used for ResNet-50 and EfficientNet-B5 have been adapted to the smaller input image size of the CIFAR10 dataset. For ResNet-50, the parameters of the first convolutional layers have parameters: stride 1, padding 1, kernel size of 3×3. No max-pooling is performed after the first convolution and global average pooling is performed over a smaller feature map of 4×4.

For EfficientNet-B5, the first convolutional layer has a stride and padding of 1 and uses a kernel size of 3×3. Similarly, the first down-sampling layer following the first convolutional layer is removed and the feature map of the global average pooling is reduced to 2×2.

For both ResNet and EfficientNet the number of output classes is changed to 10.

MLP Head Modifications For the single-layer configuration, we simply re-initialized the final FC layer. For the four-layer configuration, we extended the original three-layer MLP to have dimensions [`output`, $6Z$], [$6Z$, $3Z$], [$3Z$, Z] and [Z, #classes].

C Additional Results

Table 12: Test metrics on in-distribution data for the effect of including Focal Loss (FL) and Adaptive Focal Loss (FLA) in the training procedure. The first loss before + sign indicates the loss used for training the base model prior to TST and V-TST.

Dataset	Model	Loss	Accuracy	ECE	MCE	Train NLL	Test NLL
CIFAR10	V-TST, $m = 10$	3 FL + CE	**91.88±0.02**	**0.77±0.054**	20.60±2.488	0.0821±0.0006	0.2840±0.0011
	V-TST, $m = 1$	3 FL + CE	89.82 ± 0.07	1.52 ± 0.076	25.68 ± 8.816	0.1405 ± 0.0017	0.4124 ± 0.0023
	TST	3 FL + CE	91.86±0.01	2.03±0.020	**18.48±3.253**	**0.0652±0.0002**	**0.2577±0.0005**

D Main Results from the Authors

Table 13: Test metrics on in-distribution data test-sets from Jordahn & Olmos (2024). For V-TST, $m = 1$ and $m = 10$ indicate using 1 and 10 samples in the MC approximation, respectively. Temp. WRN is the temperature scaled WRN. Bolded fonts indicate the best-performing model for given dataset.

Dataset	Model	Accuracy	ECE	MCE	Train NLL	Test NLL
CIFAR10	V-TST, $m = 10$	**92.58±0.02**	**1.27±0.101**	28.48±9.006	0.0671±0.0004	0.2617±0.0013
	V-TST, $m = 1$	91.09±0.06	1.52±0.144	**14.77± 1.781**	0.1125±0.001	0.3732±0.0043
	TST	**92.59±0.02**	2.18±0.353	**14.89±2.249**	**0.0573±0.0002**	**0.2482±0.004**
	Temp. WRN	92.53	4.4	26.77	0.063	0.2965
	WRN	92.53	5.88	35.84	0.1038	0.4944
CIFAR100	V-TST, $m = 10$	**71.93±0.04**	**5.83±0.143**	**14.03±0.59**	0.078±0.001	1.202±0.0052
	V-TST, $m = 1$	69.18±0.09	7.34±0.291	16.85±0.694	0.1094±0.0009	1.4315±0.0104
	TST	71.26±0.06	7.07±0.084	16.74±1.942	0.0887±0.0003	**1.1331±0.002**
	Temp. WRN	71.55	15.24	33.21	**0.0739**	1.3708
	WRN	71.56	21.74	82.08	0.1161	2.2588

Table 14: Evaluation metrics on shifted data and OOD data from Jordahn & Olmos (2024).

Dataset	Model	Shift ECE	Shift MCE	OOD AUROC	OOD FPR95
CIFAR10	V-TST, $m = 10$	**10.41±0.22**	22.92±0.57	0.821±0.003	0.751±0.002
	V-TST, $m = 1$	12.89±0.3	**21.01±0.58**	0.722±0.005	0.824±0.006
	TST	11.62±0.75	25.39±1.74	**0.874±0.002**	0.699±0.009
	Temp. WRN	16.45	36.77	0.872	0.746
	WRN	20.27	45.06	**0.891**	**0.653**
CIFAR100	V-TST, $m = 10$	**14.3±0.23**	**26.96±0.46**	0.791±0.002	0.809±0.004
	V-TST, $m = 1$	16.8±0.4	30.72±0.63	0.783±0.004	0.822±0.006
	TST	17.44±0.08	28.51±0.14	**0.823±0.002**	**0.764±0.006**
	Temp. WRN	29.58	47.67	0.778	0.816
	WRN	40.47	60.38	0.785	0.892