## A. Ablation of calibration metrics

In the main paper we present calibration results computing the Expected Calibration Error with equally spaced bins, however, alternative calibration metrics have been suggested. In Fig. 9 we compare the results obtained with: ECE with equally spaced bins (ECE), ECE with equally populated bins (Ada ECE) and the Kolmogorov-Smirnov Error (KS Error). For further details see Sec. 3.3. We observe that the three different metrics yield almost identical results.

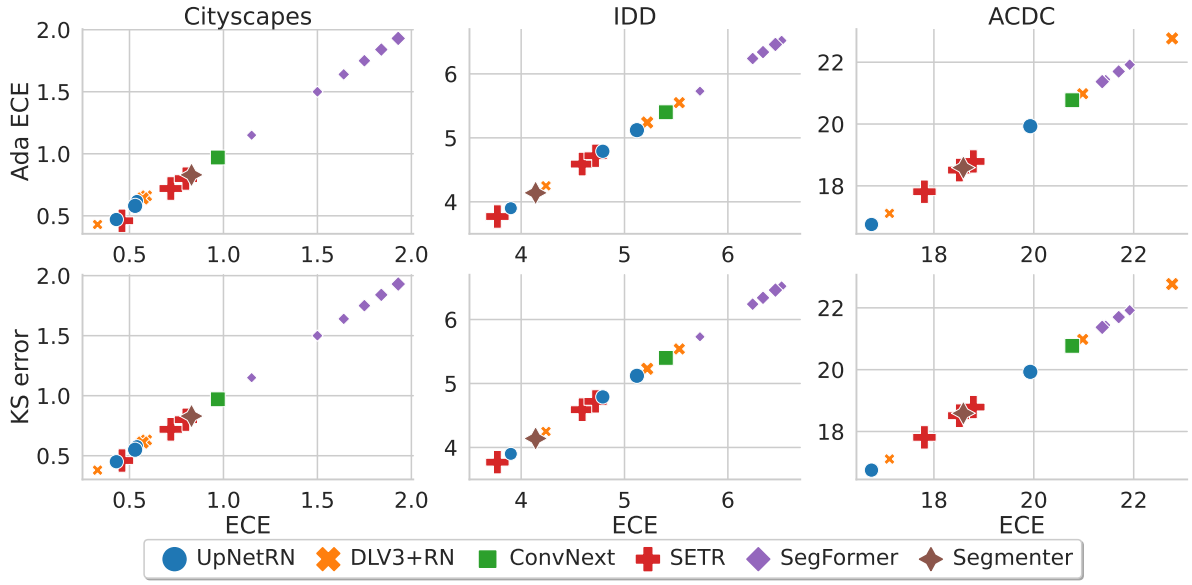


Figure 9. **Comparison of calibration error metrics** (↓) Calibration error for different datasets and networks computed with different metrics. All different metrics yield very similar result.

## B. Ablation of number of pixels for calibration

As discussed in Sec. 3.3, in segmentation, the number of samples to be taken into account for calibration scales with the number of pixels in an image. In order to be more cost-effective when testing different calibration metrics and strategies, we use a random subset of pixels within each image rather than the full image. In Fig. 10, we ablate the evolution of the different calibration metrics as we vary the number of sampled pixels. We can see that from 10k datapoints on, the metrics stabilizes; therefore, we chose to use 20k randomly sampled pixels per image for our experiments.

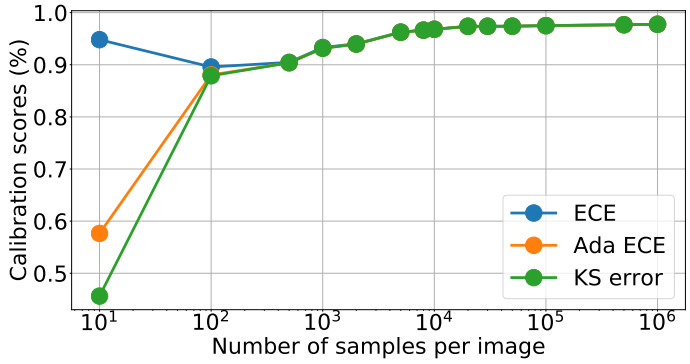


Figure 10. **Pixels-per-image ablation.** Evolution of calibration metrics as we vary the number of pixels sampled at random from each image (as opposed to the full image). We observe that when sampling more than 10k pixels all calibration metrics are very similar and the calibration error remains stable. We use 20k random samples in our experiments.

## C. Ablation of confidence score: max probability *vs*. entropy

Misclassification detection and OOD detection both rely on a metric to evaluate how confident a model is on its predictions. The most straightforward metric would be the pseudo-probability of the predicted class (i.e. the max probability). If the probability is high it is reasonable to assume that the network is confident (this is precisely what we want to impose in the calibration task). Other metrics which involve all the logits have been suggested, negative entropy being the most popular. In Fig. 11 we compare the results obtained with probabiliy and entropy as confidence metrics and observe that there is not a significant difference between the two. Therefore, the simpler confidence metric based on the predicted class probability is used for other experiments by default.

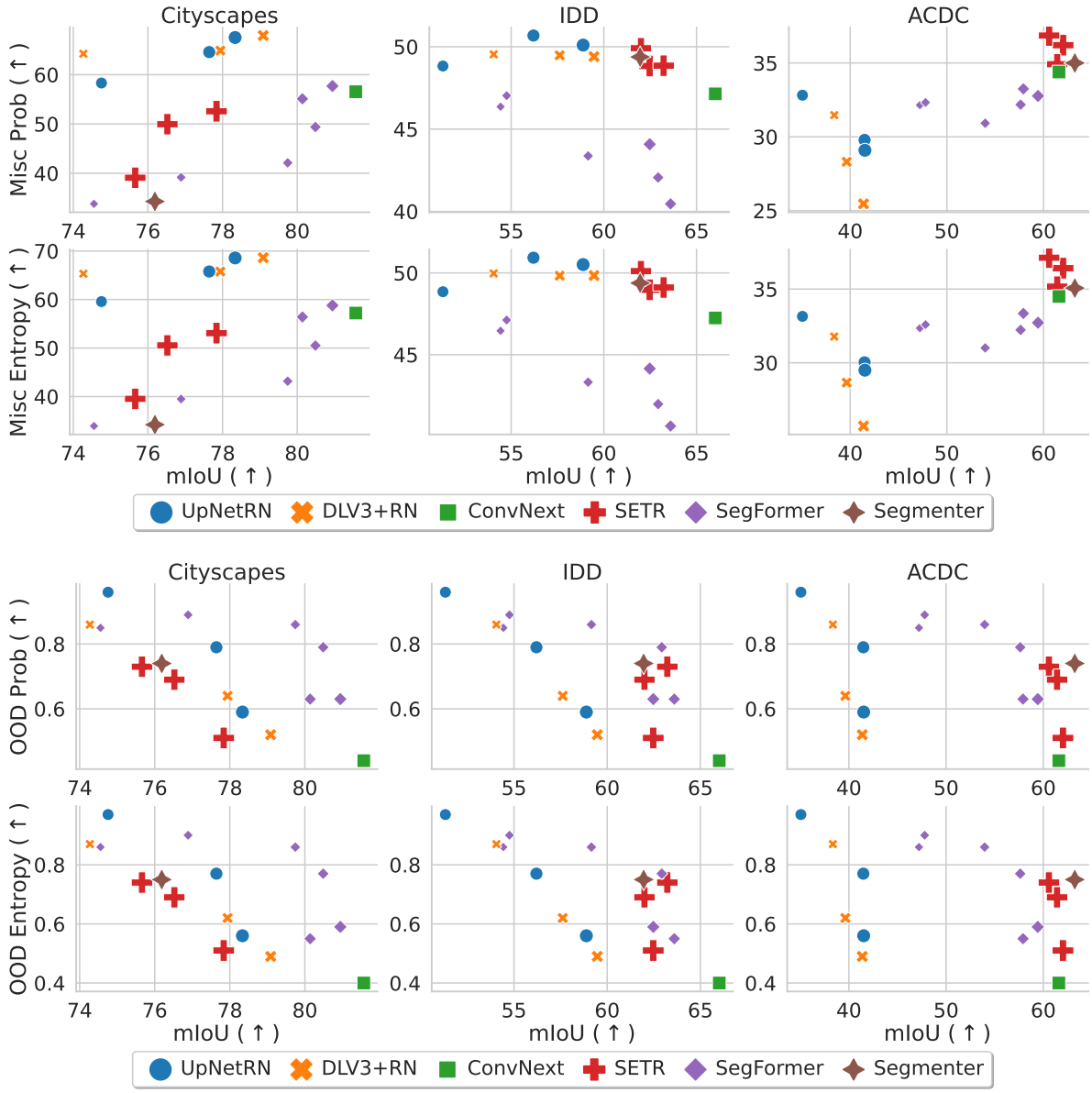


Figure 11. **Ablation of confidence score: max probability *vs*. entropy** Comparison of misclassification (top) and OOD (bottom) detection when using probability or negative entropy as confidence metrics. We observe that there is no significant difference between the two metrics, therefore we use the simpler probability as the default.

## D. Ablation number of clusters

One of the main hyperparameters in Gong *et al*. [17] is the number of clusters. In Fig. 12, we ablate the number of clusters for different test datasets (columns) and calibration datasets (rows). Although not all networks evolve in the same way, we observe that after 16 clusters, performance is more or less stable.

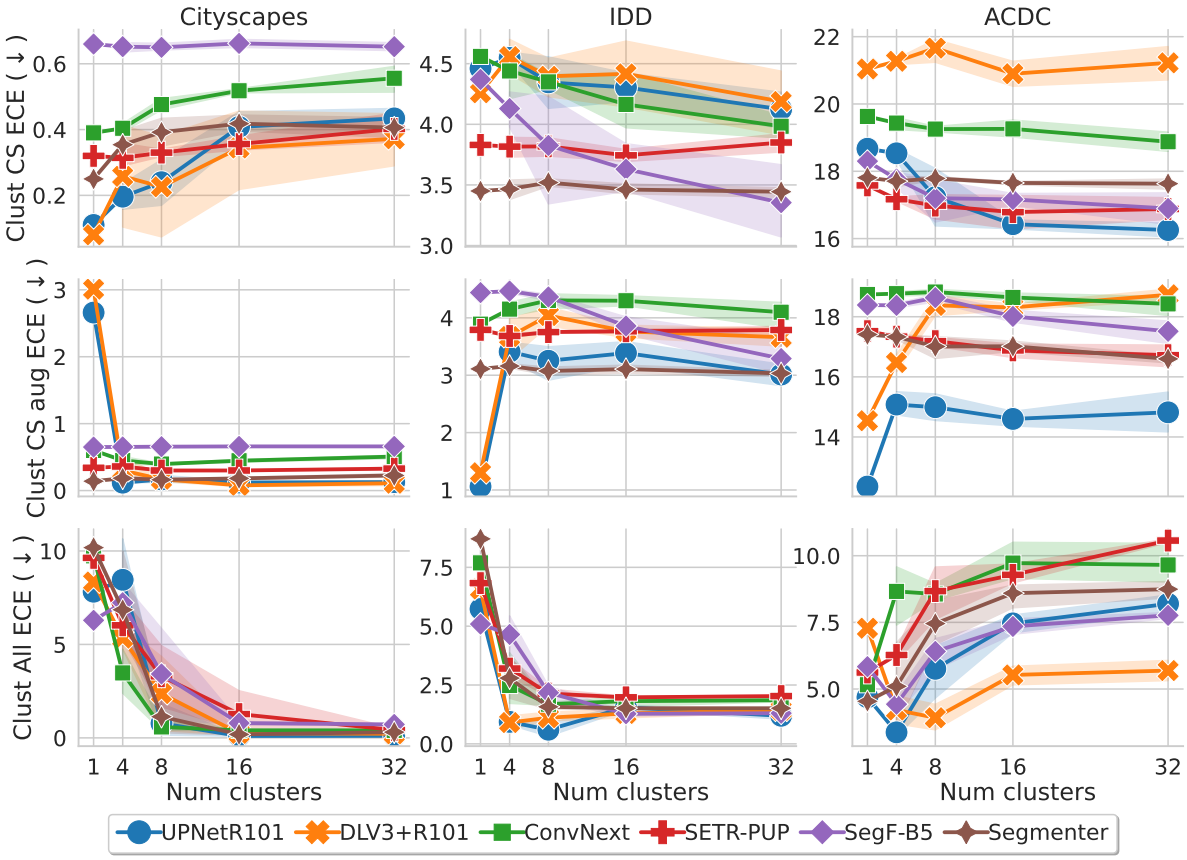


Figure 12. **Ablation number of clusters.** ECE (↓) for different models and datasets as we vary the number of clusters computed for calibration. We find 16 clusters to be relatively stable.

## E. Visualization of cluster samples

In Sec. 4.3 we observe that adaptive temperature scaling via clustering does not significantly improve calibration under distribution shift – especially when the shift is strong. The method by Gong *et al*. [17] makes the implicit assumption that the different domains captured in the clusters during calibration will be representative of the domains encountered at test time. In order to have a better intuition, we visualize a few samples randomly picked from each cluster. We show images from both the calibration set (used to compute the clusters and calibrate the models) and the test set (used to evaluate the calibration error). In our visualizations, the test set comprises images of the three datasets (CS, IDD and ACDC), while the calibration set changes for each Figure. In Figs. 13 to 15, respectively, we show representatives from clusters in *Clust All* (all datasets used during calibration), *Clust CS* (CS images used) and *Clust CS aug* (augmented CS images used). Qualitatively, when all datasets are used for calibration, the cluster assignments appear quite reasonable (*e.g*. night ACDC images are assigned to night images from calibration). However, when calibrating on CS and CS augmented, we observe that the calibration clusters are not diverse enough for the test images and the cluster assignments do not appear so intuitive.

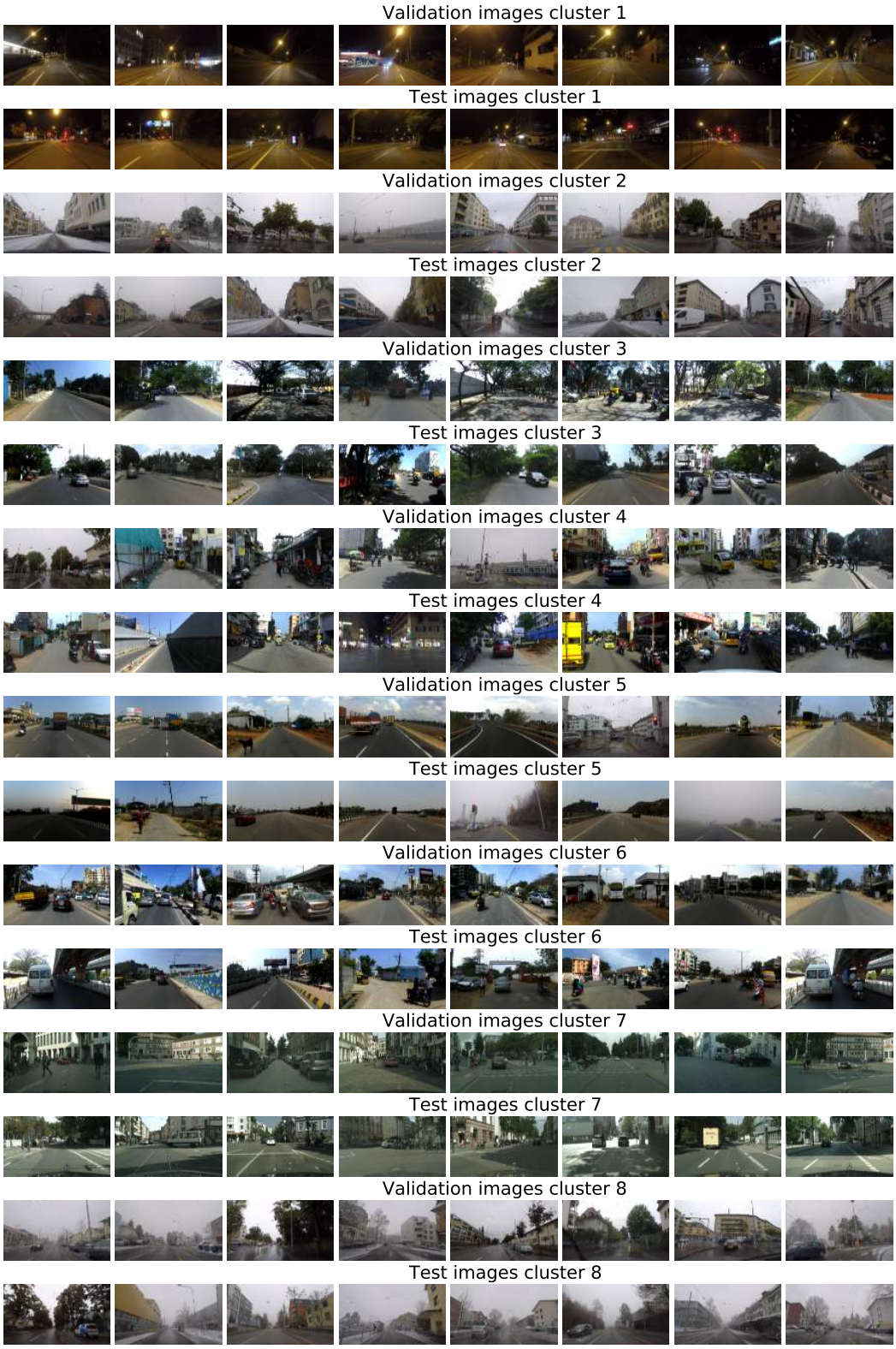


Figure 13. **Visualization of clusters (Clust All).** Sample images from clusters computed for [17]. In this case, the calibration set (where clusters are computed) contains imaged from all datasets and we qualitatively observe the cluster assignments to align with human intuition.

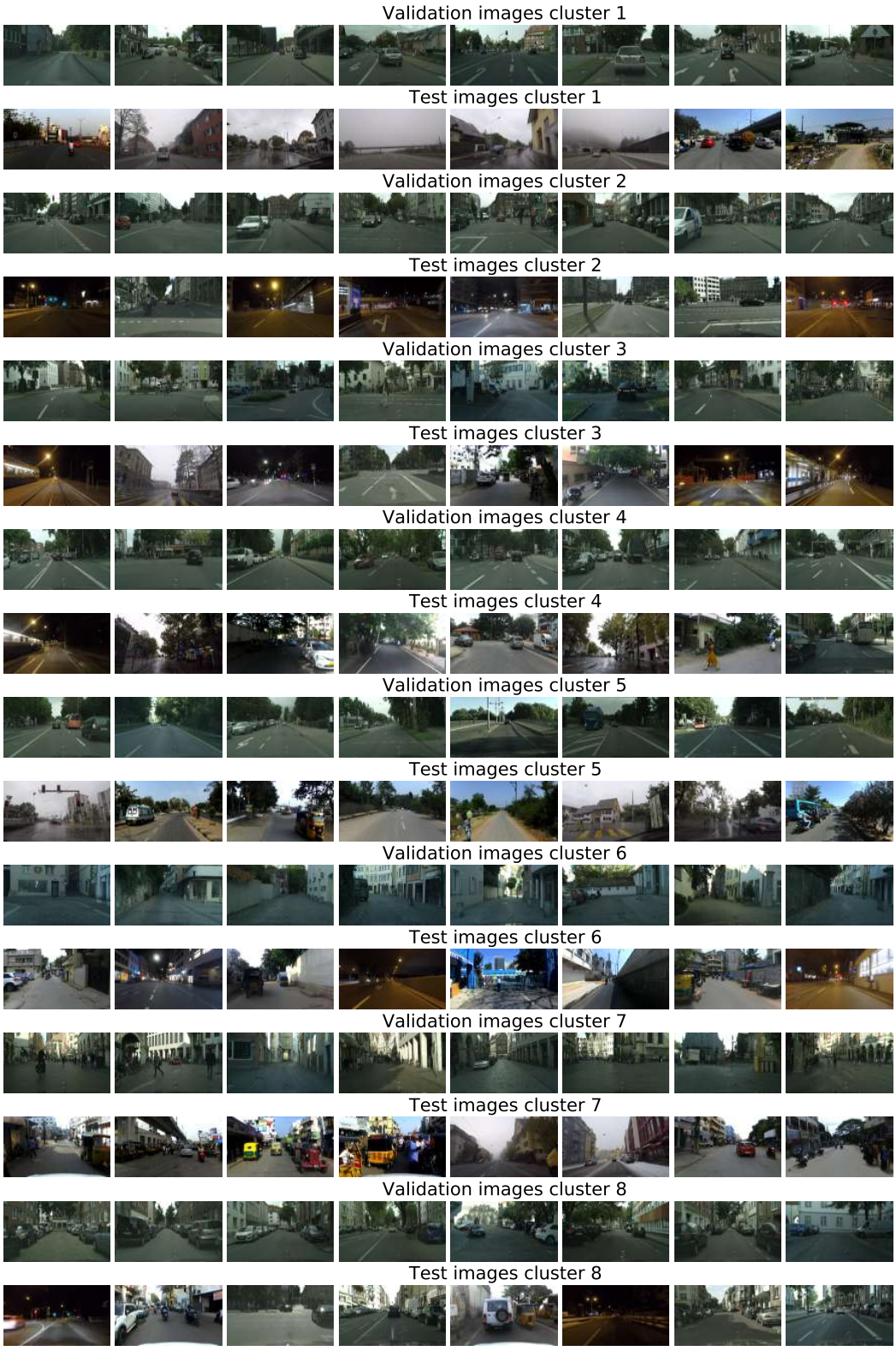


Figure 14. **Visualization of clusters (Clust CS).** Sample images from clusters computed for [17]. In this case, the calibration set (where clusters are computed) contains imaged from CS only. We qualitatively observe that the clusters are not representative of the test distribution.

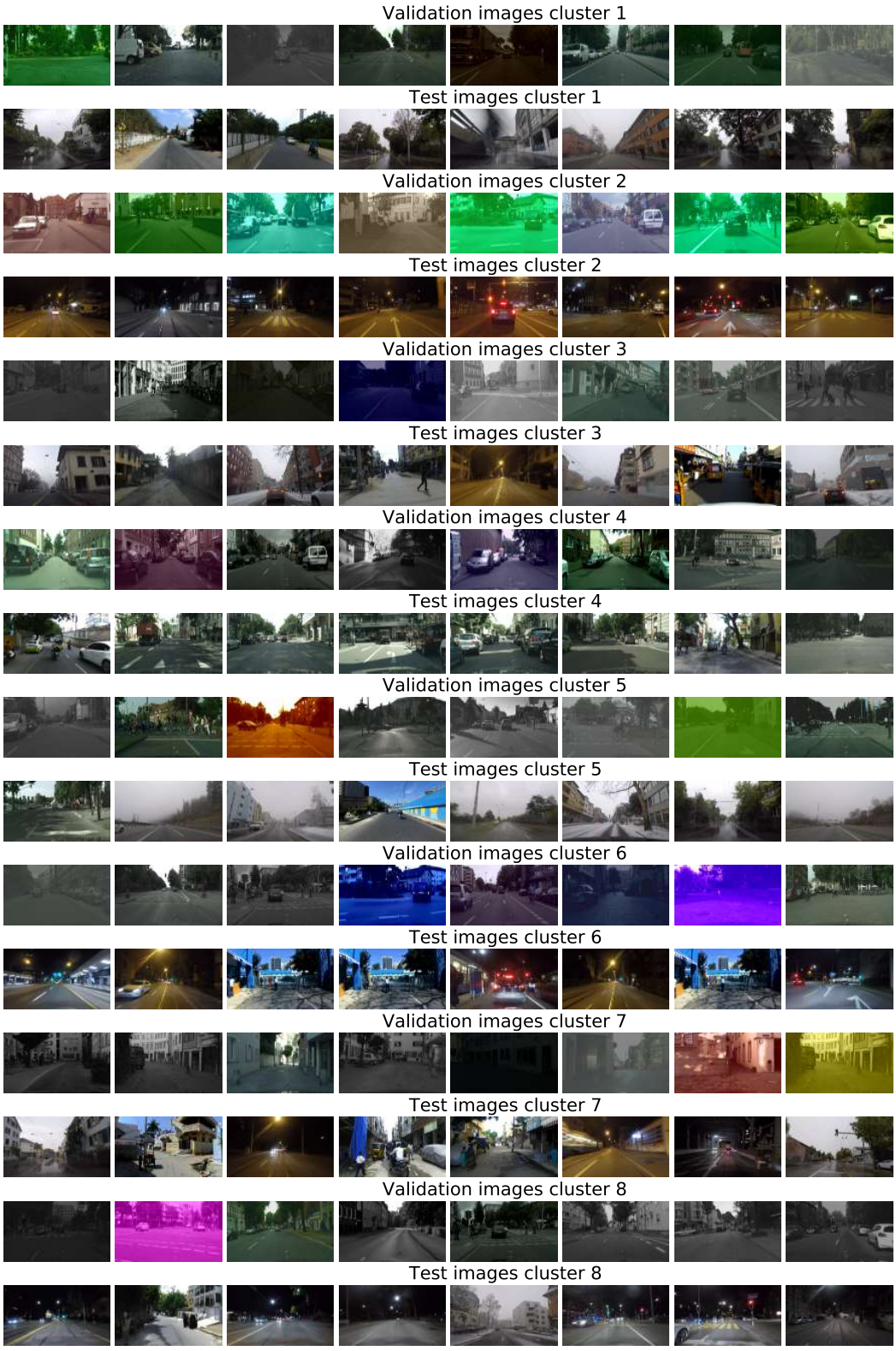


Figure 15. **Visualization of clusters (Clust CS aug).** Sample images from clusters computed for [17]. In this case, the calibration set (where clusters are computed) contains imaged from CS augmented only. Even if data augmentations introduce variability to the dataset, it is still not representative of the test distribution.

## F. Theoretical insights on adaptive temperature via clustering

In Fig. 4 we observed that even when we evaluate the ECE on the calibration set, the calibration error does not monotonically decrease as we increase the number of clusters. This is somewhat counterintuitive as one would think that, with more clusters, the temperatures can be more fine-grained and evaluating on the calibration set there are no issues with overfitting. However, it is indeed possible since the temperatures of each cluster are optimized independently. In the following we present a theorem and proof to show that decreasing the ECE for several disjoint subsets of images (clusters) independently does not guarantee that the ECE on the union set will decrease.

First, we introduce some preliminaries and notation. Consider a classifier $f : \mathcal{X} \to \mathcal{Y}$ with $\mathcal{Y} = \{1, 2, 3, \ldots, k\}$. We model our classifier as $f = \text{argmax } \hat{p}(\mathbf{y}|\mathbf{x})$ where $\hat{p}(\mathbf{y}|\mathbf{x})$ are the pseudo-probabilities estimated by the model for each class given the input. We say that a model is calibrated when

$$P\left(\mathbf{y} \mid \hat{p}(\mathbf{y}|\mathbf{x}) = p\right) = p, \quad \forall \mathbf{x} \sim \mathcal{D} \tag{1}$$

where $P$ is the true probability of the classes and $\mathcal{D}$ the data distribution. However, most works focus on a simplification of this problem where only the probability of the predicted class is taken into account, that is:

$$P\left(y = \text{argmax } \hat{p}(\mathbf{y}|\mathbf{x}) \mid \max \hat{p}(\mathbf{y}|\mathbf{x}) = p^*\right) = p^* \quad \forall \mathbf{x}, y \sim \mathcal{D}. \tag{2}$$

The most common metric to measure calibration is the *Expected Calibration Error (ECE)*. Which looks at the expected difference between the predicted and actual probabilities:

$$\mathbb{E}\Big[\ \Big|\ p^* - \mathbb{E}\left[\ P(\text{argmax } \hat{p}(\mathbf{y}|\mathbf{x}) = y) \mid \max \hat{p}(\mathbf{y}|\mathbf{x}) = p^*\ \right]\Big|\ \Big]. \tag{3}$$

In order to empirically estimate the ECE, it is standard practice to quantize the output probabilities given by the model and compute the mean probability (confidence) and accuracy in each bin. That is,

$$\widehat{\text{ECE}}_f = \sum_{i=1}^{m} \frac{\#B_i}{n} \left|\text{accuracy}(B_i) - \text{confidence}(B_i)\right| \tag{4}$$

where $\#B_i$ denotes the number of elements in the $i^{th}$ bin, $m$ denotes the number of bins and $n = \sum_{i=1}^{m} \#B_i$ the total number of elements used to estimate the ECE. We also use $f$ to indicate the dependency of the ECE on the classifier.

Consider now, that we split the data into two different sets and we quantize it in bins $\mathcal{B} = \{B_i\}$ and $\mathcal{B}' = \{B'_i\}$ with the same boundaries so that for each pair $B_i$ and $B'_i$ the range of confidence values are the same. Moreover, consider now the respective ECE computed for each subset of data independently — denoted $\widehat{\text{ECE}}_f(\mathcal{B})$ and $\widehat{\text{ECE}}_f(\mathcal{B}')$ — and on the full set of points $\widehat{\text{ECE}}_f(\mathcal{B} + \mathcal{B}')$ where $\mathcal{B} + \mathcal{B}'$ is an abuse of notation to indicate the union of elements in the bins for each index $i$.

**Theorem F.1.** *With the notation described above, consider a model $f_{oracle}$ such that an "oracle" splits the input according to whether it belongs to $\mathcal{B}$ or $\mathcal{B}'$. Moreover, $f_{oracle}$ uses two calibration strategies (one for $\mathcal{B}$ and one for $\mathcal{B}'$) in a way that it improves it's ECE on each subset $\mathcal{B}$, $\mathcal{B}'$ individually compared to some baseline model $f$ (e.g. by means of temperature scaling with a different temperature for each subset). This does not necessarily imply that the oracle model ($f_{oracle}$) will be better calibrated on the full set of points $\mathcal{B} + \mathcal{B}'$ than the baseline model $f$. That is, given that:*

$$(a) \quad \widehat{ECE}_{f_{oracle}}(\mathcal{B}) \leq \widehat{ECE}_f(\mathcal{B}) \quad \textit{and} \quad \widehat{ECE}_{f_{oracle}}(\mathcal{B}') \leq \widehat{ECE}_f(\mathcal{B}')$$

*Then, condition $(a)$ is not sufficient to claim:*

$$(b) \quad \widehat{ECE}_{f_{oracle}}(\mathcal{B} + \mathcal{B}') \leq \widehat{ECE}_f(\mathcal{B} + \mathcal{B}')$$

*Proof.* In order to proof the theorem we will construct a counter-example where condition $(a)$ is satisfied but condition $(b)$ is not. Consider the accuracy and confidence for the full set of points in a given bin:

$$\text{acc}(B_i + B'_i) = \frac{\#B_i\ \text{acc}(B_i) + \#B'_i\ \text{acc}(B'_i)}{\#B_i + \#B'_i}$$
$$\text{conf}(B_i + B'_i) = \frac{\#B_i\ \text{conf}(B_i) + \#B'_i\ \text{conf}(B'_i)}{\#B_i + \#B'_i}$$

Then the ECE of the full set of points will be:

$$\begin{aligned}\widehat{\mathrm{ECE}}_f(\mathcal{B}+\mathcal{B}') &= \sum_{i=1}^{m} \frac{\#B_i + \#B'_i}{n+n'} \left|\mathrm{acc}_f(B_i+B'_i) - \mathrm{conf}_f(B_i+B'_i)\right| \\ &= \sum_{i=1}^{m} \frac{1}{n+n'} \left| \#B_i(\mathrm{acc}_f(B_i) - \mathrm{conf}_f(B_i)) + \#B'_i(\mathrm{acc}_f(B'_i) - \mathrm{conf}_f(B'_i))\right|.\end{aligned}$$

To simplify the notation, let us define $r_i(f) = \mathrm{acc}_f(B_i) - \mathrm{conf}_f(B_i)$ and similarly for $r'_i(f)$. Then, we can write:

$$\begin{aligned}\widehat{\mathrm{ECE}}_f(\mathcal{B}+\mathcal{B}') &= \sum_{i=1}^{m} \frac{1}{n+n'} |\#B_i\ r_i(f) + \#B'_i\ r'_i(f)|. \\ \widehat{\mathrm{ECE}}_f(\mathcal{B}) &= \sum_{i=1}^{m} \frac{\#B_i}{n} |r_i(f)|. \\ \widehat{\mathrm{ECE}}_f(\mathcal{B}') &= \sum_{i=1}^{m} \frac{\#B'_i}{n'} |r'_i(f)|.\end{aligned}$$

Now let us consider a setting where $r_i(f) = r$ and $r_i(f_{\text{oracle}}) = -0.5r$ for some $r \neq 0$ while $r'_i(f) = -r$ and $r'_i(f_{\text{oracle}}) = -0.5r$. Moreover, consider $\#B_i = \#B'_i$ which implies $n = n'$, then this setting would satisfy condition $(a)$ since

$$\begin{aligned}\widehat{\mathrm{ECE}}_f(\mathcal{B}) = \sum_{i=1}^{m} \frac{\#B_i}{n} |r| \geq \sum_{i=1}^{m} \frac{\#B_i}{n} |-0.5r| = \widehat{\mathrm{ECE}}_{f_{\text{oracle}}}(\mathcal{B}) \quad \text{and} \\ \widehat{\mathrm{ECE}}_f(\mathcal{B}') = \sum_{i=1}^{m} \frac{\#B'_i}{n'} |-r| \geq \sum_{i=1}^{m} \frac{\#B'_i}{n'} |-0.5r| = \widehat{\mathrm{ECE}}_{f_{\text{oracle}}}(\mathcal{B}').\end{aligned}$$

However, this same setting would not satisfy condition $(b)$ since

$$\begin{aligned}\widehat{\mathrm{ECE}}_f(\mathcal{B}+\mathcal{B}') = \sum_{i=1}^{m} \frac{\#B_i}{2n} |r-r| = 0 \quad \text{and} \\ \widehat{\mathrm{ECE}}_{f_{\text{oracle}}}(\mathcal{B}+\mathcal{B}') = \sum_{i=1}^{m} \frac{\#B_i}{2n} |-0.5r-0.5r| > 0.\end{aligned}$$

Thus, we have showed that condition $(a)$ does not imply $(b)$. □

This result implies that minimizing the ECE for different subsets of the data independently (e.g. each cluster of images) does not necessarily lead to an overall improvement of the ECE. Moreover, we have assumed only two sets of samples without loss of generalization since if $(a)$ implied $(b)$ for an arbitrary number of data splits it would in particular imply it for two. Finally, note that our result is valid for either image classifiers or segmentors. In the first case we would each prediction would be the class of a whole image while in the second case the each pixel in an image would have a different prediction.

## G. Per-class clustering

In Fig. 6, we have observed that adaptive temperature via clustering [17] does not significantly help improving out-of-domain calibration compared to local temperature scaling (LTS) [14]. One important difference between the methods is that LTS computes a temperature for each pixel in the image while clustering is performed at the image level – using a single temperature per image. This motivates us to perform an ablation where, on top of the image level clustering, pixels in a given image are grouped according to their predicted class. Intuitively, we are looking for a temperature for regions in the image that look alike to the network (since they are assigned to the same class). In Fig. 16 we compare standard per-image clustering (top) with the aforementioned per-class clustering (bottom). Note that per-class clustering always groups pixels according to the predicted class, therefore if $k = 1$ then there are 19 clusters (corresponding to the CS classes). Calibration images are from the CS dataset.

Similarly to per-image clustering, increasing the number of clusters does not seem to always help when using per-class clustering. Moreover, we do not find that per-class clustering significantly improves calibration except for SegFormer architecture. We are not stating here that finer-grained clustering may not yield further improvements (and reach similar performance to LTS). However, given that improving ECE in different subdomains independently is not guaranteed to improve overall calibration (see Appendix F), perhaps a different approach to finding the temperatures and clusters taking into account both local and global calibration error would be needed.

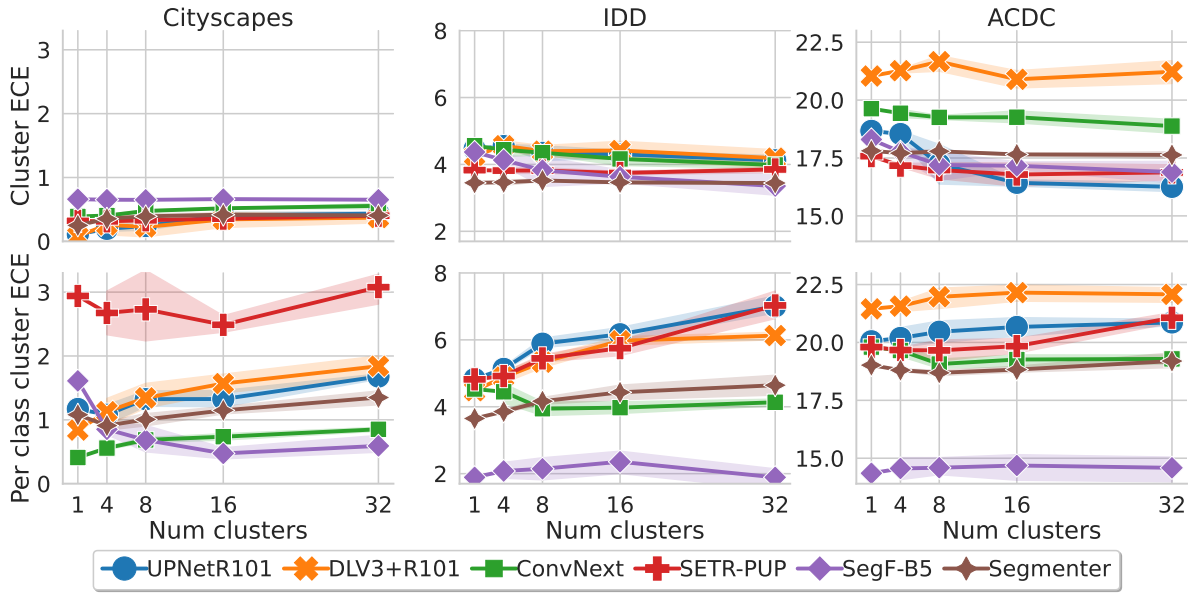


Figure 16. **Per-class clustering ablation.** ECE (↓) for different models and datasets as we vary the number of clusters computed for calibration. We compare per-image clustering (where all pixels in an image are given the same temperature) *vs*. per-class clustering (where different pixels in an image are given a temperature depending on their predicted class). Overall, we do not observe consistent improvements when using per-class clustering except in SegFormer architecture.

## H. Ablation LTS: image *vs*. logits

LTS [14] employs a small-weight calibration network which receives both the image and predicted logits as input and it returns a temperature map to scale the logits (with a different temperature for each pixel in the image). Given its remarkable performance (see Figs. 5 and 6) and, to get further insights into this method, in Fig. 17 we perform an ablation where the calibration network only receives the image or the logits as input. To carry out this experiment, we modify the network in [14] so that both input branches (logits and image) receive the same input, either both logits or both image. All calibration networks have been trained in CS images only.

Interestingly, we observe that, in distribution (ECE CS), the better performing method for most networks is the LTS variant that uses the logits only. On the other hand, for OOD calibration, the better performing variant in most cases is the one that relies on both logit and image information. Moreover, under strong domain shifts (ECE ACDC), LTS yields better results by using only the image information than by using only the logits. However, this is subject to variability as results vary across different architectures. Further investigations on how logit and image signals are combined may constitute a promising direction to further improve calibration results.

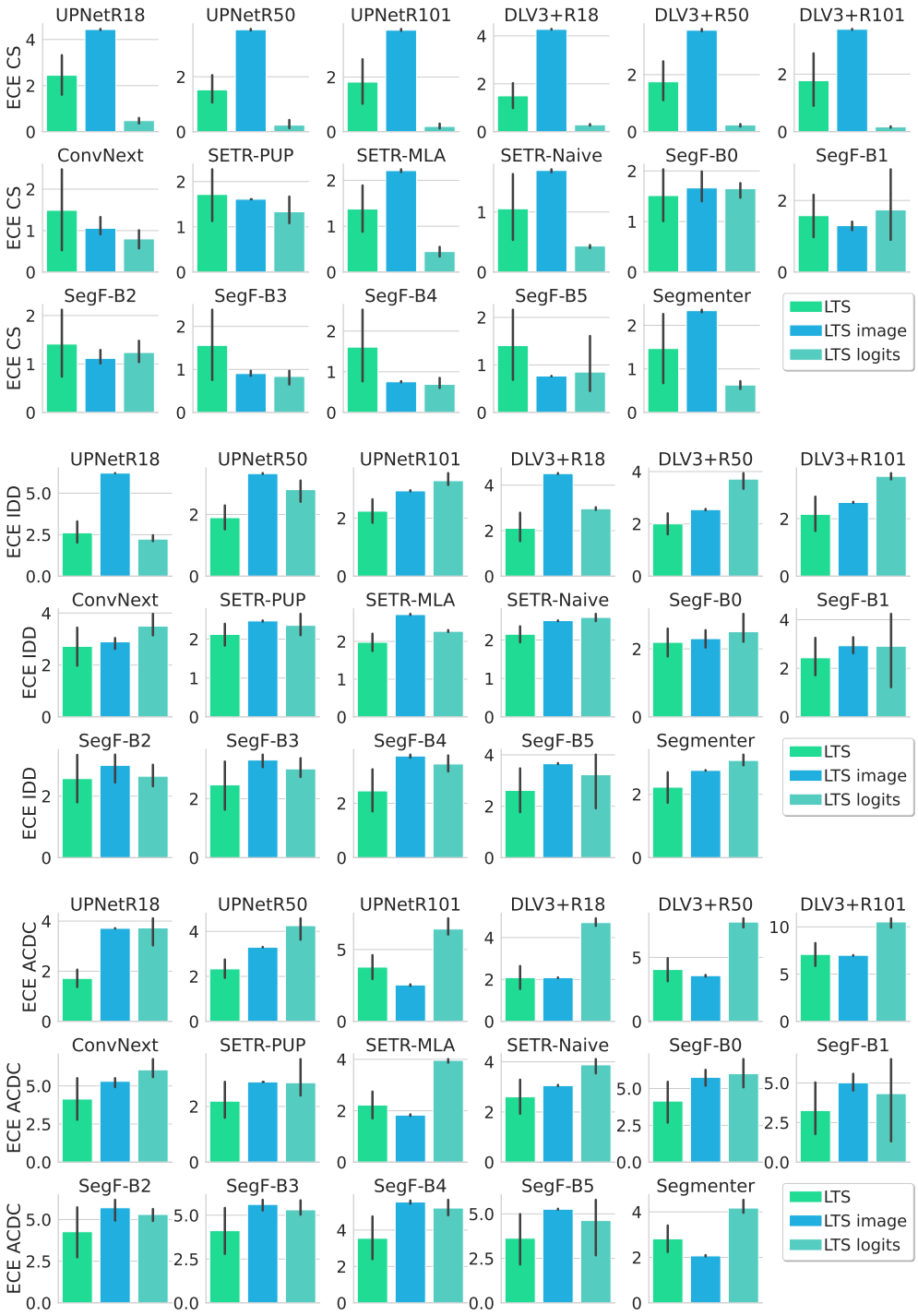


Figure 17. **LTS with image *vs*. logits information** ECE (↓) after calibration with three LTS variants: the original method combining image and logits (LTS), an LTS version only using logit information (LTS logits) and the complementary version only using image information (LTS image). We observe that in distribution, using only the logits seems to perform better while out of distribution using information from both image and logits works best.

## I. Ablation of the effect of training settings

Our goal is to assess the reliability of the best available models for segmentation — the ones used by practitioners. Therefore, to test a model at its best, we need to use its most favorable setup (original pre-training, optimizer, etc.), as done in previous work [33, 44]. Nevertheless, in this section we ablate the effects of two different training settings: the dataset used to pre-train the backbone and the number of training iterations. Most recent models were pre-trained on ImageNet 21k, however, ResNet backbones are usually pre-trained on ImageNet 1k. In the top rows of Tab. 2 we compare the performance of a BiT-ResNet50 [26] and a ConvNext-B [30] backbones pre-trained with either ImageNet 1k or 21k. We observe that the benefits of a larger pre-training dataset are small in comparison to the gap between architectures.

On the other hand, most models were trained using 80k iterations (as is quite standard for CS dataset, but Segmenter and SegFormer models' original training schedule uses 160k iterations. In the bottom rows of Tab. 2 we compare Segmenter and SegFormer-B5 networks trained with either 160k or 80k iterations, again the differences are minor compared to the gap between architectures.

| | **mIoU** (↑) | | | **ECE** (↑) | | |
|---|---|---|---|---|---|---|
| **Architecture** | **CS** | **IDD** | **ACDC** | **CS** | **IDD** | **ACDC** |
| ConvNext-B (IN-1k) | 80.39 | 64.11 | 58.77 | 0.77 | 5.53 | 18.95 |
| ConvNext-B (IN-21k) | 81.56 | 65.70 | 60.59 | 0.81 | 5.27 | 20.07 |
| BiT-RN50 (IN-1k) | 76.36 | 57.46 | 47.47 | 0.67 | 5.18 | 19.47 |
| BiT-RN50 (IN-21k) | 76.49 | 57.23 | 46.70 | 0.65 | 5.20 | 19.63 |
| Segmenter (160k) | 76.19 | 61.96 | 63.24 | 0.83 | 4.14 | 18.59 |
| Segmenter (80k) | 76.22 | 60.74 | 62.96 | 0.71 | 4.28 | 18.36 |
| SegF-B5 (160k) | 80.94 | 62.47 | 59.43 | 1.93 | 6.46 | 21.37 |
| SegF-B5 (80k) | 80.15 | 62.95 | 57.00 | 1.07 | 5.32 | 21.27 |

Table 2. Ablations of different training settings. On the top we compare ConvNext and BiT-RN50 architectures with the backbones pre-trained on either ImageNet 1k or 21k datasets. We observe that the benefits of a larger pre-training dataset are small in comparison to the gap between architectures. On the bottom we compare the amount of training iterations, again, we observe that although longer training schedules do improve the performance, changes are also small in comparison to architecture differences.

## J. Architectures for Universal Image Segmentation

With the appearance of transformers, and in particular motivated by DETR [4], some architectures have been proposed with the objective to solve the three main Image Segmentation tasks, that is: Semantic Segmentation, Instance Segmentation and Panoptic Segmentation [8, 9, 57]. Here we will focus on Mask2Former [8] which is based on MaskFormer [9] and is the best performing universal architecture to the best of our knowledge. Interestingly, to be able to solve the different segmentation tasks jointly, these architectures do not output the standard per-pixel logits when it comes to semantic segmentation. Instead, they predict a set of $N$ object masks (where $N$ is fixed) and the class probabilities for each object. Then, to obtain the per-pixel class probabilities they marginalize over all the possible objects a pixel could belong to, we refer the reader to [8, 9] for further details.

Although this final output can be regarded as per-pixel class probabilities, they way it is obtained differ from the standard *logits + softmax* setting that all calibration methods rely on, therefore it is not straightforward to compare these universal architectures with other models with temperature scaling. Nevertheless, given the good performance and wider applicability of these models, we include them in our study comparing only off-the-shelf performance in terms of mIoU, calibration, OOD detection and misclassification. The best performing model from [8] is based on a Swin transformer (Swin Large) [29], therefore, we also include a Swin transformer model with UpperNet to the comparison for completeness.

### J.1. Calibration with Mask2Former

In 18 we present the ECE vs segmentation error (100 - mIoU) for the different models. Interestingly, we observe that Mask2Former seems to be the best-performing model in terms of mIoU in all datasets. In terms of calibration error Mask2Former is poorly calibrated in distribution (CS) but has a milder increase in ECE as the distribution shift becomes stronger.

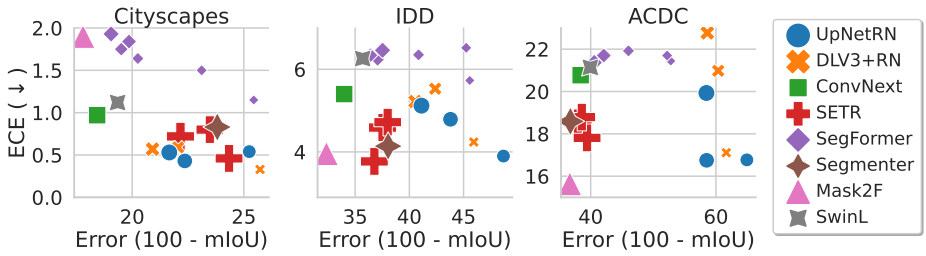


Figure 18. **mIoU error** (↓) ***vs*. Expected calibration error** (↓). All models were trained on CS. Markersize proportional to number of parameters. Interestingly, we observe that Mask2Former seems to be the best-performing model in terms of mIoU in all datasets. In terms of calibration error Mask2Former is poorly calibrated in distribution (CS) but has a milder increase in ECE as the distribution shift becomes stronger.

## J.2. OOD detection with Mask2Former

In 19 we present the OOD vs mIoU for the different models. Here we observe that Mask2Former and Swin transformer align with the negative trend observed in previous models where models with better mIoU tend to perform worse at OOD detection.

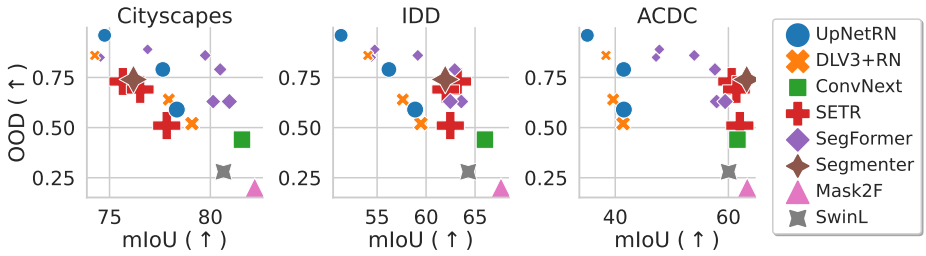


Figure 19. **mIoU** (↑) ***vs*. OOD - AUROC** (↑) for different model families. All models trained on CS. Markersize proportional to number of parameters. We observe that Mask2Former and Swin transformer align with the negative trend observed in previous models where models with better mIoU tend to perform worse at OOD detection.

## J.3. PRR with Mask2Former

In 20 we present the OOD vs mIoU for the different models. Although Mask2Former is significantly better than other models in terms of misclassification detection in distribution, it seems to perform significantly worse under strong domain shifts (ACDC). This seems to be contrary to the trend followed by other models where robustness seems to be correlated with misclassification under domain shift.

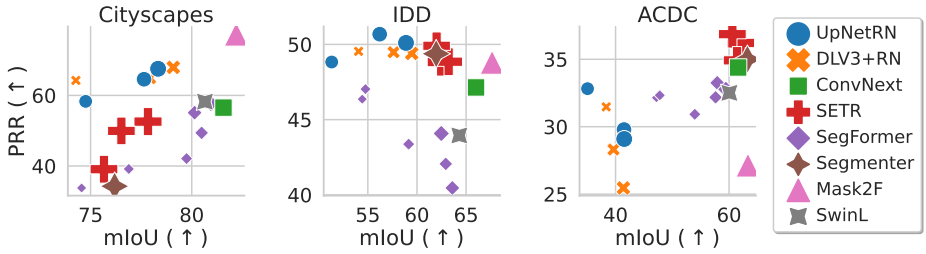


Figure 20. **mIoU** (↑) ***vs*. prediction rejection ratio** (↑) for different model families. All models trained on CS. Markersize proportional to number of parameters. Mask2Former is significantly better than other models in terms of misclassification detection in distribution, it seems to perform significantly worse under strong domain shifts (ACDC).

## K. Visualization of uncertainty and temperature maps

As opposed to classification, semantic segmentation models can assign different confidence to regions of the image. That can allow, among other applications, to detect regions with low confidence that may correspond to novel classes or weird instances of a known class. ACDC shares the same classes as Cityscapes, however, IDD includes some novel classes which are not included in the Cityscapes dataset. In 21 we illustrate some examples of IDD images with confidence maps before and after LTS, together with the temperature scaling maps predicted by the calibration network. We observe how different OOD classes (*e.g*. autorickshaw, bridge, billboard) or weird instances are highlighted in the calibration maps. In Appendix L we quantify how useful are the calibration maps in order to perform local OOD detection.

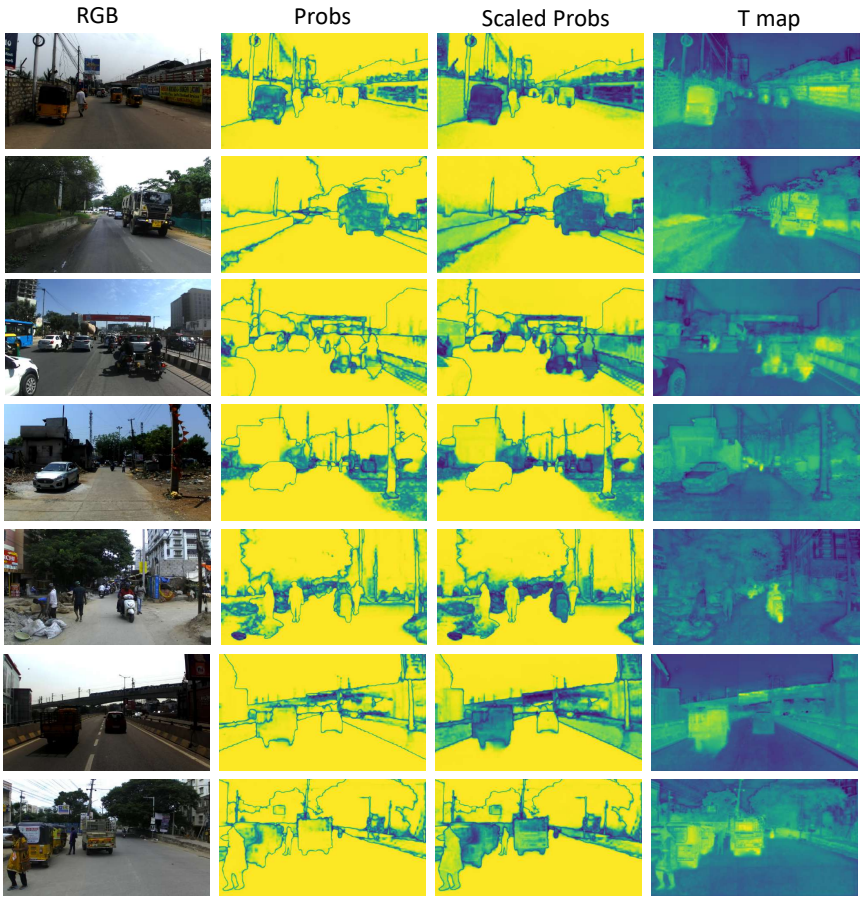


Figure 21. **Calibration and temperature maps** for different IDD images. We observe how different OOD classes (*e.g*. autorickshaw, bridge, billboard) or weird instances are highlighted in the calibration maps.

## L. Local OOD detection

In this section we perform OOD detection at the pixel level to find regions of the image that belong to unknown classes (autorickshaw, guardrail, billboard, bridge). We define pixels of unknown classes as OOD while those corresponding to CS classes are in distribution. In Fig. 22 we present the results of local OOD detection vs mIoU. We make two observations: i) Mask2Former has the best local OOD detection; ii) Differences between models are smaller for local OOD: numbers are roughly in the $0.7 - 0.8$ range *vs*. $0.4 - 1.0$ range for image-based OOD.

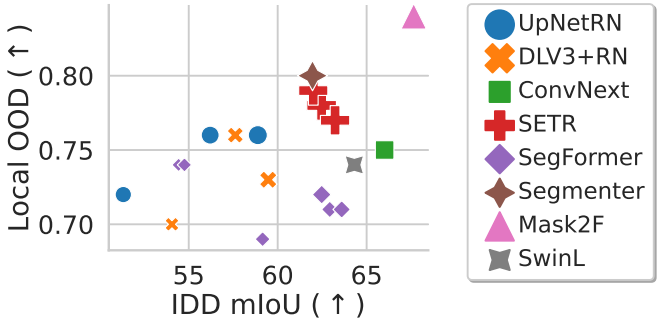


Figure 22. **mIoU** (↑) ***vs*. local ood** (↑) for different model families. All models trained on CS. Markersize proportional to number of parameters. Mask2Former achieves the best local OOD detection performance.

## M. Additional plots: adaptive temperature via clustering

In Fig. 3 we analyzed the calibration error after applying the method by Gong *et al*. [17], which clusters the images in the calibration set and computes a different temperature per cluster. Due to space constraints we only showed the best performing model for each family, in Fig. 23 we show the results for all models.


Figure 23. **ECE (↓) after clustering TS** Extension of Fig. 3 in the main paper where we show results for all models.

Complementary to Fig. 23 where we show the different calibration methods for a given architecture with barplots, in Fig. 24 we present the same results but we group them by calibration method (instead of by model). For each test dataset (rows), we plot the ECE *vs*. mIoU after calibrating the models with the corresponding method (columns).

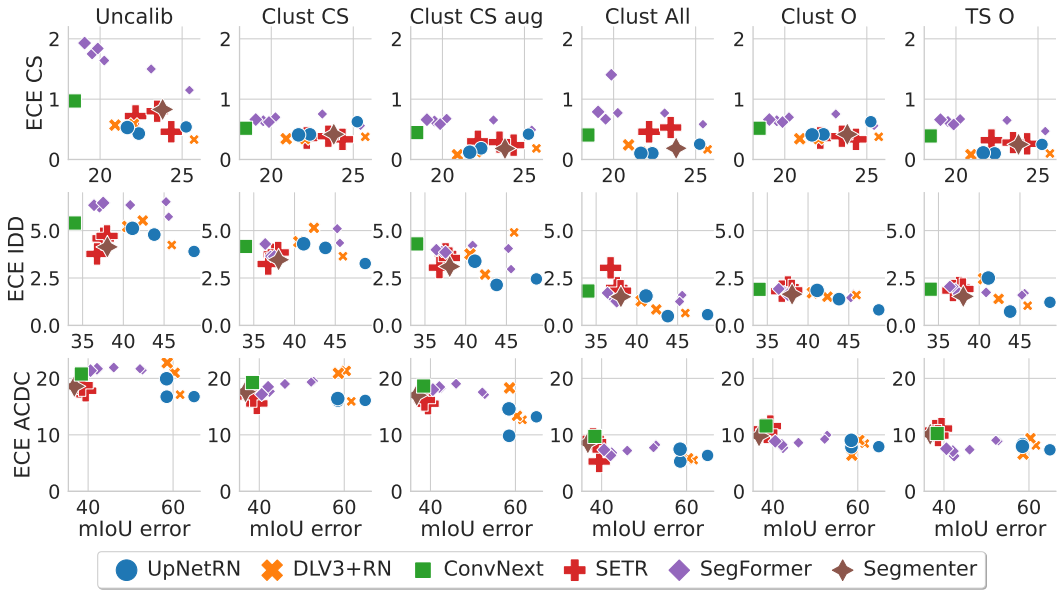


Figure 24. **ECE** (↓) ***vs*. mIoU error** (↓) **after calibration with clustering TS** considering different calibration sets. This plot provides a different visualization of the results in Fig. 23.

## N. Additional plots: local temperature scaling

In Fig. 5 we analyzed the calibration error after applying the method by Ding *et al*. [14] which learns a calibration network that predicts the temperature as a function of the image and segmentation model logits. Due to space constraints we only showed the best performing model for each family, in Fig. 25 we show the results for all models.


Figure 25. **ECE (↓) after Local TS** Extension of Fig. 5 in the main paper where we show results for all models.

Complementary to Fig. 25 where we show the different calibration methods for a given architecture with barplots, in Fig. 26 we present the same results but we group them by calibration method (instead of by model). For each test dataset (rows), we plot the ECE *vs*. mIoU after calibrating the models with the corresponding method (columns).

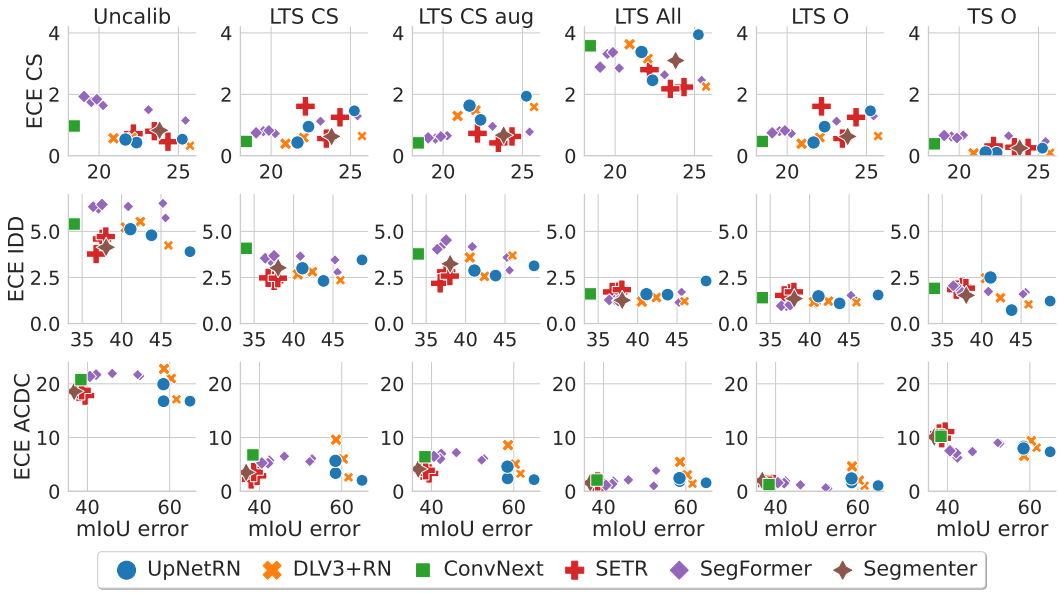


Figure 26. **ECE** (↓) ***vs*. mIoU error** (↓) **after calibration with Local TS** considering different calibration sets. This plot is showing the same results as Fig. 25 but in a different visualization.

## O. Additional plots: Comparison calibration methods

In Fig. 6 we compared the calibration error after calibrating with TS CS, Clust CS and LTS CS *vs*. the uncalibrated baseline. Due to space constraints we only showed the best performing model for each family, in Fig. 27 we show the results for all models.

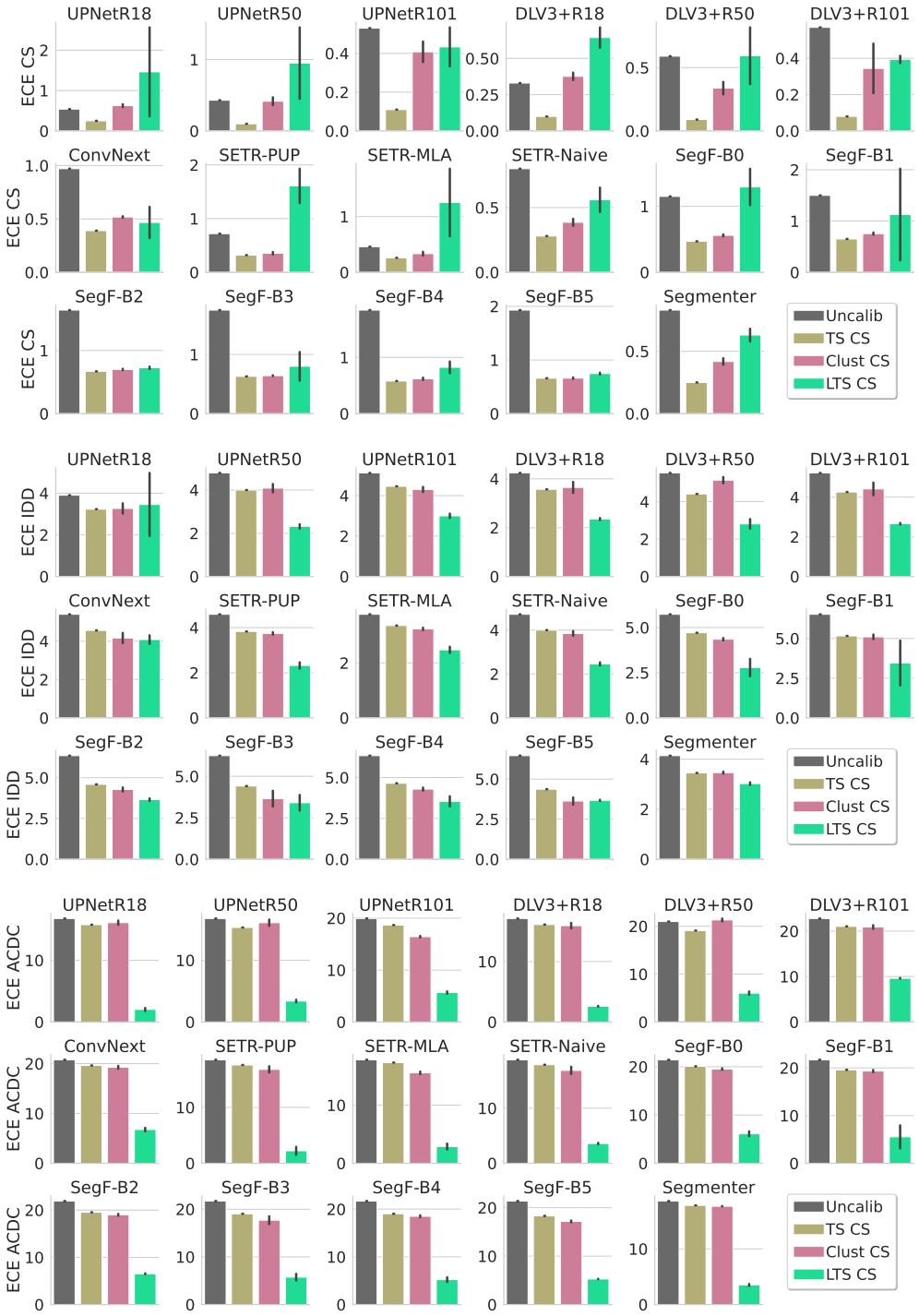


Figure 27. **ECE (↓) after calibration with different methods** Extension of Fig. 6 in the main paper where we show results for all models.

Complementary to Fig. 27 where we show the different calibration methods for a given architecture with barplots, in Fig. 28 we present the same results but we group them by calibration method (instead of by model). For each test dataset (rows), we plot the ECE *vs*. mIoU after calibrating the models with the corresponding method (columns).

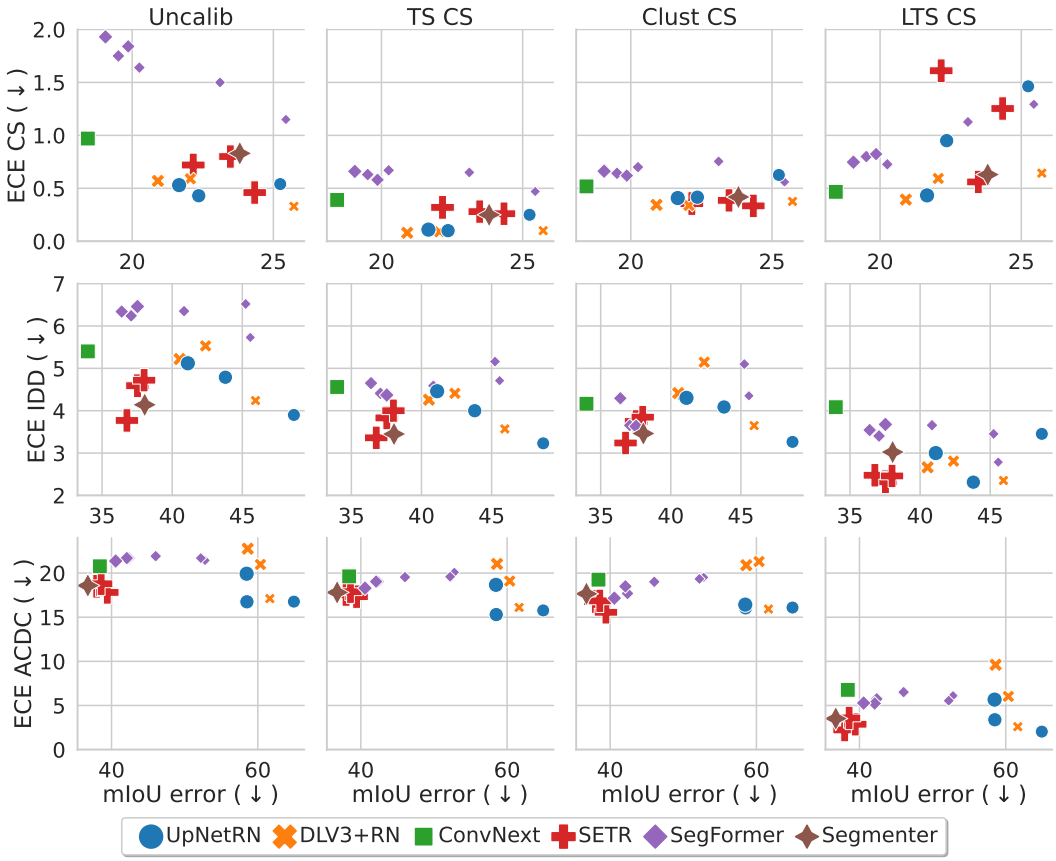


Figure 28. **ECE** (↓) ***vs*. mIoU error** (↓) **after calibration with different methods** on the CS calibration set. This plot is showing the same results as Fig. 27 but in a different visualization.

## P. Additional plots: misclassification detection

In Fig. 8 we compared the misclassification detection and OOD detection performance of the networks after calibrating with TS CS, Clust CS and LTS CS *vs*. the uncalibrated baseline. Due to space constraints we only showed the best performing model for each family, in Fig. 29 we show the misclassification detection results for all models.

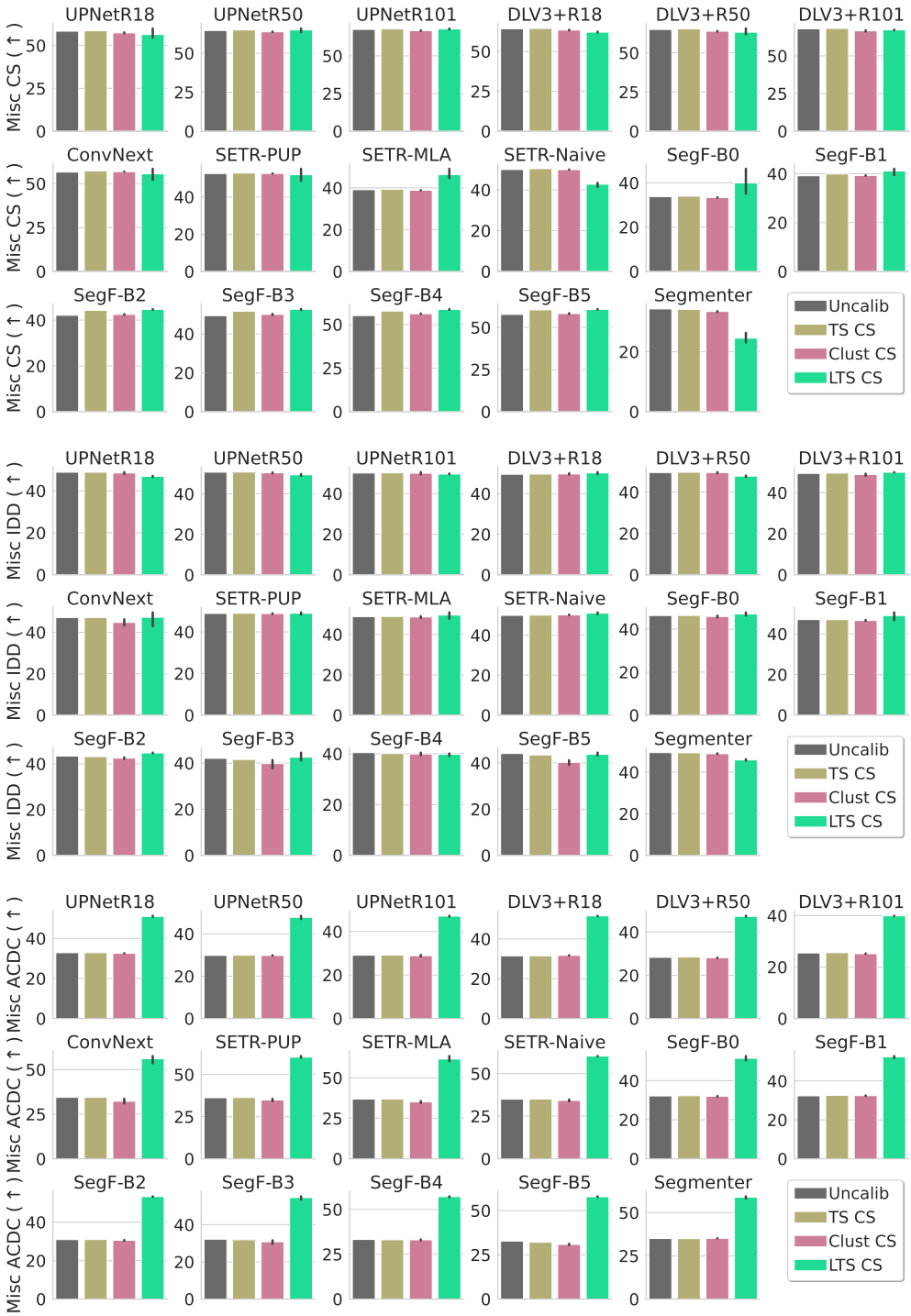


Figure 29. **Misc detection: PRR (↑) after calibration with different methods** Extension of Fig. 8 in the main paper where we show misclassification results for all models.

Complementary to Fig. 29 where we show the different calibration methods for a given architecture with barplots, in Fig. 30 we present the same results but we group them by calibration method (instead of by model). For each test dataset (rows), we plot the ECE *vs*. mIoU after calibrating the models with the corresponding method (columns).

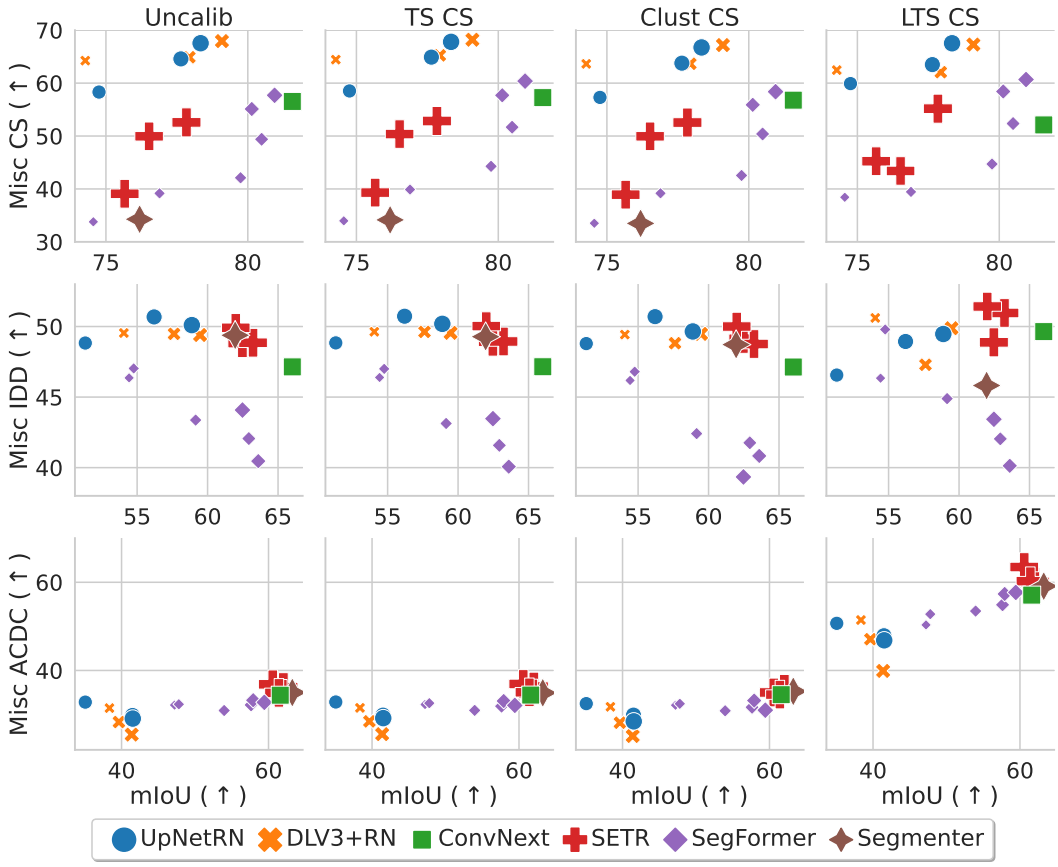


Figure 30. **PRR** (↑) ***vs*. mIoU** (↑) **after calibration with different methods** on the CS calibration set. This plot is showing the same results as Fig. 29 but in a different visualization.

## Q. Additional plots: out-of-distribution detection

In Fig. 8 we compared the misclassification detection and ood detection performance of the networks after calibrating with TS CS, Clust CS and LTS CS *vs*. the uncalibrated baseline. Due to space constraints we only showed the best performing model for each family, in Fig. 29 we show the OOD detection results for all models.

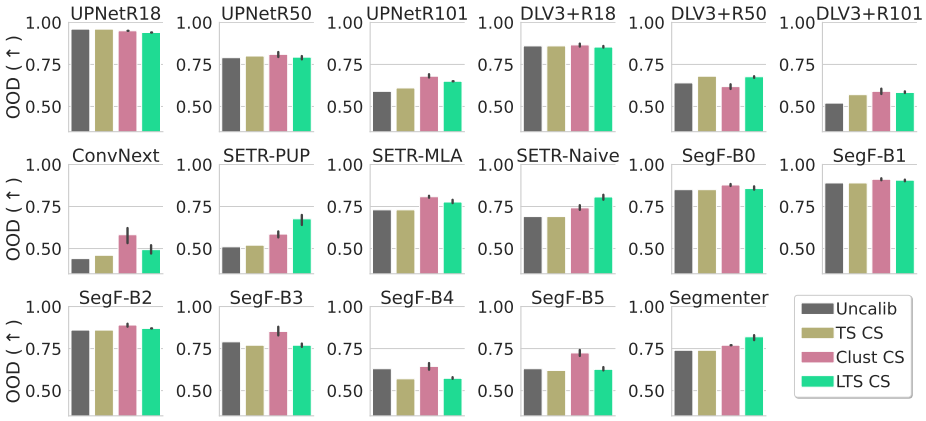


Figure 31. **OOD detection: AUROC (↑) after calibration with different methods** Extension of Fig. 8 in the main paper where we show OOD detection results for all models.

Complementary to Fig. 31 where we show the different calibration methods for a given architecture with barplots, in Fig. 32 we present the same results but we group them by calibration method (instead of by model). For each test dataset (rows), we plot the ECE *vs*. mIoU after calibrating the models with the corresponding method (columns).

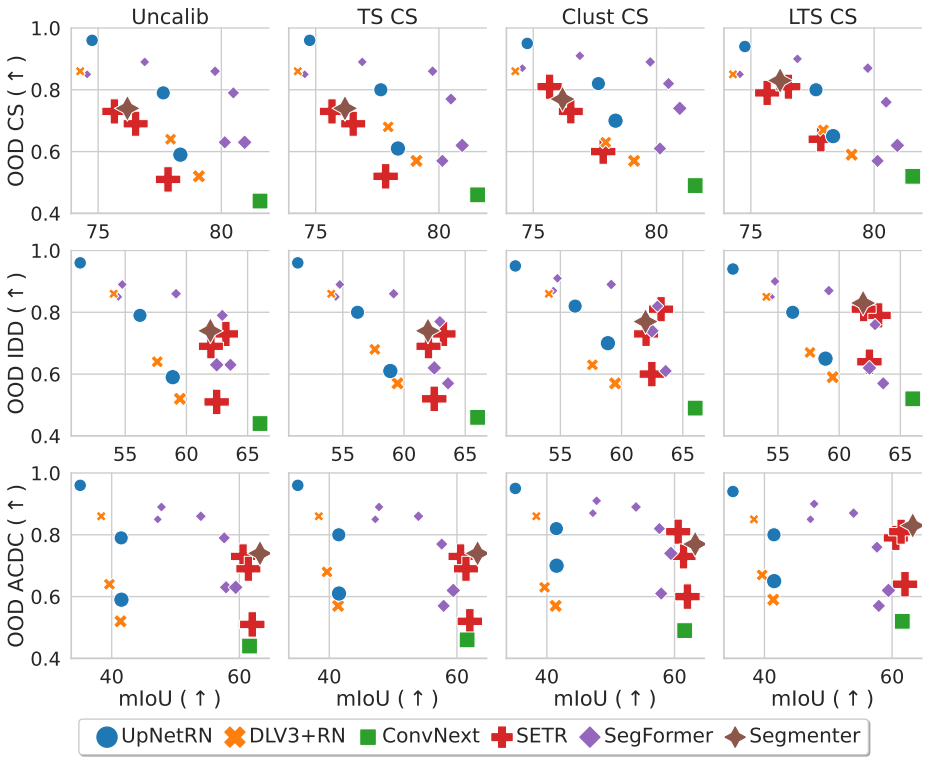


Figure 32. **AUROC (↑) *vs*. mIoU (↑) after calibration with different methods** on the CS calibration set. This plot is showing the same results as Fig. 31 but in a different visualization.