Class and Region-Adaptive Constraints for Network Calibration

Balamurali Murugesan, Julio Silva-Rodriguez, Ismail Ben Ayed, and Jose Dolz

ETS Montreal

Abstract. In this work, we present a novel approach to calibrate segmentation networks that considers the inherent challenges posed by different categories and object regions. In particular, we present a formulation that integrates class and region-wise constraints into the learning objective, with multiple penalty weights to account for class and region differences. Finding the optimal penalty weights manually, however, might be unfeasible, and potentially hinder the optimization process. To overcome this limitation, we propose an approach based on Class and Region-Adaptive constraints (CRaC), which allows to learn the class and region-wise penalty weights during training. CRaC is based on a general Augmented Lagrangian method, a well-established technique in constrained optimization. Experimental results on two popular segmentation benchmarks, and two well-known segmentation networks, demonstrate the superiority of CRaC compared to existing approaches. The code is available at: https://github.com/Bala93/CRac/

Keywords: Network calibration · Uncertainty · Segmentation.

1 Introduction

Despite the remarkable progress achieved by deep neural networks (DNNs), they are susceptible to suffer from miscalibration, leading to overconfident predictions [7,19], even when they are incorrect. This issue becomes especially significant in safety-critical scenarios, such as medical diagnosis or treatment, where producing accurate uncertainty estimates is of paramount importance. An inherent cause of network miscalibration is known to be the implicit bias for low-entropy predictions caused by popular supervised losses, such as the cross-entropy, which encourages large differences between the logit of the ground truth category and the remaining classes [20].

A myriad of approaches have emerged to mitigate network miscalibration, which mainly focus on either post-processing strategies or integrating additional learning objectives during training. The first family of approaches, i.e., *post-processing* methods, offers a simple alternative for modifying the softmax predictions in a post-hoc fashion by establishing a mapping from raw network outputs to well-calibrated confidences [5,7,8,34,35]. The second category involves incorporating additional regularization during training, typically penalizing low-entropy predictions. For example, [28] introduced an explicit term that maximizes the Shannon entropy of the network predictions during training, which was later extended in [13] by penalizing low-entropy distributions only in incorrect predictions. Furthermore, popular losses for classification, such as Label smoothing [32] or focal loss [14], implicitly integrate an entropy maximization term, which has a favourable effect on calibration [20,21]. More recently, [15,16,23] propose to enforce inequality constraints on the logit space, allowing to control the margin on logit distances, ultimately reducing overconfidence in the predictions. This provided more flexibility than systematically maximizing the entropy of the predictions, as in [20,21], which results in gradients that continually push towards a non-informative solutions. Other works include the integration of pair-wise constraints between classes [4] or augmenting the training dataset by convex combinations of random pairs of images and their associated labels, e.g., MixUp [33]. Nevertheless, even though these works have achieved remarkable progress in addressing miscalibration in both classification [4,7,8,20,21,28,34] and segmentation tasks [5,13,15,23], they disregard neighbour pixel relationships, in terms of classes, which is of significant relevance in semantic image segmentation.

Certainly, one of the factors contributing to the reduced performance of these losses in segmentation tasks arises from the uniform, or near-to-uniform, distribution enforced in the network predictions (whether logit or softmax predictions), which neglects the spatial context [22]. To overcome this issue, and to integrate class-wise information of the surrounding pixels during training, Spatially Varying Label Smoothing (SVLS) [10] introduced a label smoothing strategy that captures the structural uncertainty required in semantic segmentation. More specifically, SVLS uses a Gaussian kernel applied across the one-hot encoded ground truth, leading to class probabilities based on a soft combination of neighboring pixels. As exposed in [22,24], SVLS integrates an implicit penalty on softmax predictions, which enforces a prior based on soft class proportions of surrounding pixels. This strategy, however, lacks a mechanism to control the influence of the constraint over the main objective, potentially hindering the optimization process. To circumvent this limitation, authors presented a simple solution that combines the standard cross-entropy with an explicit penalty, where both the prior and its impact can be easily controlled.

Although the work proposed in [22,24] achieves greater calibration performance than existing alternatives, and integrates class-relationships across a pixel and its neighbours, it presents two major limitations: *1)* The scalar balancing weight that controls the importance of the penalty is equal for all classes, and for all the regions. This scenario is suboptimal, as it can hamper the network performance when some classes are more challenging to segment, or under-represented. Furthermore, this strategy considers than the weight of the penalty should be the same for a pixel inside the object (likely to have *low uncertainty*) than for a pixel within the organ boundaries (likely to have *high uncertainty*). *2)* The value of the balancing weight is defined before network optimization, lacking an adaptive strategy during training. For example, as the training evolves, the cross-entropy loss pushes towards lower-entropy predictions, whereas the penalty weight is the same at the beginning and the end of the training. Based on these findings, we can summarize our contributions as:

1. We propose a class and region-adaptive constraint approach to tackle the miscalibration issue in semantic segmentation models. In particular, we formulate a solution that considers the specificities of each category and different regions by introducing independent class and region-wise penalty weights. This contrasts with the prior work in [22], where a uniform scalar penalty weight is employed, regardless of categories or regions.
2. Furthermore, we transfer the constrained problem to its dual unconstrained optimization counterpart by using an Augmented Lagrangian method (ALM). This alleviates the need for manually adjusting each penalty weight and allows, through a series of iterative *inner* and *outer* steps, to find the optimal value of each penalty weight, which can be learned in an adaptive manner.
3. Comprehensive experiments on two popular segmentation benchmarks, and with two well-known segmentation backbones, demonstrate the superiority of our approach over a set of relevant recent calibration approaches.

2 Methodology

Notation. We denote the training dataset as $\mathcal{D}(\mathcal{X}, \mathcal{Y}) = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$, where $\mathbf{x}^{(n)} \in \mathcal{X} \subset \mathbb{R}^{\Omega_n}$ represents the n^{th} image, Ω_n its spatial image domain, and $\mathbf{y}^{(n)} \in \mathcal{Y} \subset \mathbb{R}^{K}$ the corresponding pixel-wise ground-truth annotation with K classes, which is provided as a one-hot encoding vector. Given an input image $\mathbf{x}^{(n)}$, a neural network parameterized by $\boldsymbol{\theta}$ generates a logit vector $f_{\boldsymbol{\theta}}(\mathbf{x}^{(n)}) = \mathbf{l}^{(n)} \in \mathbb{R}^{\Omega_n \times K}$, which can be converted into probability values with the softmax operator, $\text{softmax}(\mathbf{l}^{(n)}) = \mathbf{s}^{(n)} \in [0,1]^{\Omega_n \times K}$. To simplify the notations, we omit sample indices, as this does not lead to any ambiguity.

2.1 Background

Despite its importance in dense prediction tasks, such as segmentation, very few approaches consider pixel spatial relationships across classes to address the miscalibration issue. Spatially Varying Label Smoothing (SVLS) [10] integrates neighbour class information by softening the pixel label assignments with a discrete spatial Gaussian kernel. More recently, NACL [22,24] formally showed that SVLS actually enforces an implicit constraint on soft class proportions of surrounding pixels, and propose the following constrained optimization problem:

$$\min_{\boldsymbol{\theta}} \quad \mathcal{L}_{CE} \quad \text{s.t.} \quad \boldsymbol{\tau} = \mathbf{l}, \tag{1}$$

which can be approximated by incorporating an explicit penalty, whose overall learning objective is defined as:

$$\min_{\boldsymbol{\theta}} \quad \sum_{i \in \Omega} \sum_{k \in K} (-y_k^{(i)} \log(s_k^{(i)}) + \lambda |\tau_k^{(i)} - l_k^{(i)}|). \tag{2}$$

The first term in the above equation is the standard cross-entropy loss on a given pixel, the second term is a linear penalty over the pixel logit distributions, $\boldsymbol{\tau}$ is a prior, and λ the balancing hyperparameter that controls the importance of each term. With this objective, when the constraint $|\tau_k - l_k|$ deviates from 0 (i.e., τ_k and l_k are different) the value of the penalty term increases. Thus, as the prior $\boldsymbol{\tau} = \{\tau_0, ..., \tau_{K-1}\}$ captures the class distribution of a 2D patch[1] surrounding the pixel, the penalty enforces the predicted logit distribution $\mathbf{l}$ to follow $\boldsymbol{\tau}$.

2.2 Class and region-wise penalties

The unconstrained formulation presented in Equation 2 employs a single uniform penalty. We argue that this scenario is suboptimal, as it disregards differences across individual categories, or even different regions with different uncertainty in the target object, which may pose distinct inherent learning challenges. For example, annotations from a patch in the center of an organ typically have less uncertainty that labels in within the organ boundaries. A better, and more optimal strategy would integrate multiple penalty weights λ, one for each category and type of patch/region, leading to a set of penalty weights $\boldsymbol{\Lambda} \in \mathbb{R}_{+}^{K\times R}$, with R being the number of regions. For simplicity, in this work we will consider only two types of regions (i.e., $R = 2$, leading to $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1\}$), that we denote as *inner* and *outer* regions, and whose sets are defined as $\mathcal{I}$ and $\mathcal{O}$, respectively. More concretely, if the surrounding ground truth patch of a given pixel only contains one category, it will be considered as an *inner* patch, whereas otherwise it will be an *outer* patch. Thus, we can formally define our formulation as:

$$\min_{\theta} \sum_{i\in\Omega} \mathcal{H}(\mathbf{y}^{(i)}, \mathbf{s}^{(i)}) + \sum_{i\in\mathcal{I}} \sum_{k\in K} \lambda_{k,0} |\tau_k^{(i)} - l_k^{(i)}| + \sum_{i\in\mathcal{O}} \sum_{k\in K} \lambda_{k,1} |\tau_k^{(i)} - l_k^{(i)}|, \quad (3)$$

where $\mathcal{H}(\mathbf{y}, \mathbf{s})$ is the standard cross-entropy loss. As stated in prior literature in constrained convolutional neural networks [18,30,16,31], while $\boldsymbol{\Lambda}^* \in \mathbb{R}_{+}^{K\times R}$ are the Lagrange multipliers of the presented problem, and $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^*$ could be considered the best choice to solve (3), using $\boldsymbol{\Lambda}^*$ as the penalty weights may not feasible in practice. On the other hand, finding the optimal value for each penalty weight manually can pose optimization challenges, particularly for datasets with a large number of classes.

2.3 The proposed class and region adaptive solution

General Augmented Lagrangian. To alleviate the need of having to chose the penalty weights $\boldsymbol{\Lambda} \in \mathbb{R}_{+}^{K\times R}$, we propose to use an Augmented Lagrangian Multiplier (ALM) method. ALM approaches are optimization techniques that integrate penalties and primal-dual updates to efficiently tackle constrained optimization problems. These methods iteratively refine solutions by adjusting

[1] More details about the priors and the enforced constraint in [22,24].

penalty terms based on Lagrange multipliers, effectively balancing between satisfying constraints, i.e., the penalties, and minimizing the main objective function, in our case the cross-entropy loss. ALM approaches are favoured due to their ability to handle complex constraints and their robust performance across various optimization scenarios, and enjoy widespread popularity in the general context of optimization [2,27]. A general constrained optimization problem can be formally defined as:

$$\min_{x} \quad g(x) \quad \text{s.t.} \quad h_i(x) \leq 0, \quad i = 1, \ldots, n \tag{4}$$

with $g : \mathbb{R}^d \to \mathbb{R}$ the *objective function* and $h_i : \mathbb{R}^d \to \mathbb{R}, i = 1, \ldots, n$ being the *set of constraint functions*. Generally, this problem is tackled by solving a succession of $j \in \mathbb{N}$ unconstrained problems, each solved approximately w.r.t x:

$$\min_{x,\lambda} \quad \mathcal{L}^{(j)}(x) = g(x) + \sum_{i=1}^{n} P(h_i(x), \rho_i^{(j)}, \lambda_i^{(j)}), \tag{5}$$

where $P : \mathbb{R} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \to \mathbb{R}$ is a *penalty-Lagrangian function*, whose derivative w.r.t. its first variable $P'(z, \rho, \lambda) \equiv \frac{\partial}{\partial z} P(z, \rho, \lambda)$ exists, is positive and continuous for all $z \in \mathbb{R}$ and $(\rho, \lambda) \in (\mathbb{R}_{++})^2$. In addition, we denote $\boldsymbol{\rho}^{(j)} = (\rho_i^{(j)})_{1 \leq i \leq n} \in \mathbb{R}_{++}^n$ and $\boldsymbol{\lambda}^{(j)} = (\lambda_i^{(j)})_{1 \leq i \leq n} \in \mathbb{R}_{++}^n$ as the penalty parameters and multipliers associated to the penalty P at the iteration j. We detail in the Appendix the set of axioms that any penalty function P must satisfy [3].

The ALM can be split into two iterations. First, in the *outer* iterations, which indexed by j, the *penalty multipliers* $\boldsymbol{\lambda}$ and the *penalty parameters* $\boldsymbol{\rho}$ are updated. Then, during the *inner* iterations, the objective $\mathcal{L}^{(j)}$ (Eq. 5) is minimized using the previous solution as initialization to this problem. Particularly, the penalty multipliers $\boldsymbol{\lambda}^{(j)}$ are updated to the derivative of P w.r.t. to the solution obtained during the last *inner* step:

$$\lambda_i^{(j+1)} = P'(h_i(x), \rho_i^{(j)}, \lambda_i^{(j)}). \tag{6}$$

This approach increases the value of the penalty multipliers when the constraint is violated, and decreases their value otherwise. Thus, integrating an ALM during optimization enables an *adaptive* and *learnable* strategy to determine an optimal value for the penalty weights.

Our global learning objective. Based on the benefits detailed above, we propose to solve the problem in Eq. 3 by using an ALM approach. More concretely, we reformulate this problem by integrating a penalty function P, which is parameterized by $(\boldsymbol{\rho}, \boldsymbol{\lambda}) \in \mathbb{R}_{++}^K \times \mathbb{R}_{++}^K$:

$$\begin{aligned} \min_{\theta, \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1} \quad & \sum_{i \in \Omega} \mathcal{H}(\mathbf{y}^{(i)}, \mathbf{s}^{(i)}) + \sum_{i \in \mathcal{I}} \sum_{k \in K} P(\tau_k^{(i)} - l_k^{(i)}, \rho_{k,0}, \lambda_{k,0}) \\ & + \sum_{i \in \mathcal{O}} \sum_{k \in K} P(\tau_k^{(i)} - l_k^{(i)}, \rho_{k,1}, \lambda_{k,1}). \end{aligned} \tag{7}$$

To obtain an accurate estimate of the penalty multipliers at each epoch, we compute the satisfaction of the constraint on the validation set, following standard practices in machine learning. In this work, we consider that a single training epoch approximately minimizes the loss function. Then, we compute the average penalty multiplier on the validation set. This means that, after a training epoch j, the penalty multipliers for all $k = 0, ..., K-1$ and each region r at epoch $j+1$ can be computed as:

$$\lambda_{k,r}^{(j+1)} = \frac{1}{|\mathcal{D}_{val}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{val}} P'\left(\tau_k - l_k, \rho_{k,r}^{(j)}, \lambda_{k,r}^{(j)}\right). \tag{8}$$

Furthermore, ρ is updated as:

$$\rho_{k,r}^{(j+1)} = \begin{cases} \gamma \rho_{k,r}^{(j)} & |\tau_k^{(i)} - l_k^{(i)}| > \mu * |\tau_k^{(i)} - l_k^{(i)}|; \\ \rho_{k,r}^{(j)} & \text{otherwise,} \end{cases} \tag{9}$$

where μ is a constant factor that determines the amount of the update. Last, following prior works on ALM in the context of constrained CNNs [16,30,31], we employ PHR as the penalty, which is defined as:

$$\text{PHR}(z, \rho, \lambda) = \begin{cases} \lambda z + \frac{1}{2}\rho z^2 & \text{if} \quad \lambda + \rho z \geq 0; \\ -\frac{\lambda^2}{2\rho} & \text{otherwise.} \end{cases} \tag{10}$$

3 Experiments

Datasets. Following the Neighbor Aware Calibration Loss (NACL) [22,24], we use the ACDC and FLARE datasets with its setting. **ACDC** [1] contains 100 patient exams with cardiac MR volumes and their respective pixel-wise annotations. We follow the standard practices on this dataset, and extract 2D slices from the volumes, which are resized to 224×224. Furthermore, **FLARE** [17] includes 360 volumes of multiple organs in abdominal CTs, together with their corresponding segmentation masks, which are resampled to a common space and cropped to 192×192×30.

Baselines. We compare to relevant calibration losses, as well as to state-of-the-art methods for calibration in medical image segmentation: Focal Loss (FL) [20], penalizing low-entropies (ECP) [28], Label smoothing (LS) [32], SVLS [10], MbLS [15], NACL [22] and BWCR [11]. As segmentation backbones, we have selected two well-known and popular networks, U-Net [29] and nnU-Net [9].

Implementation details. For most of the compared methods, we use the hyperparameters values reported in [22]: FL ($\gamma = 3$), LS ($\alpha = 0.1$), ECP ($\lambda = 0.1$), MbLS ($\lambda = 0.1$ and $m = 10$), SVLS ($\sigma = 2$) and NACL ($\lambda = 0.1$). Furthermore, for BWCR, the impact of the logit consistency is controlled by $\lambda_{min} = 0.01$, and $\lambda_{max} = 1$. Regarding the prior used in NACL and our method CRaC, we use the one proposed in [22], defined as $\tau_k = \sum_{i=1}^{d} y_i^k$, and which is computed over a 3×3 patch. We train all the models during 100 epochs, with ADAM [12] as optimizer and a batch size fixed to 16. The learning rate is set to 10^{-3} for the

Table 1: **Quantitative performance.** Discriminative (DSC ↑, HD ↓) and calibration (ECE ↓, TACE ↓) metrics, using U-Net as segmentation backbone. The best method is highlighted in bold, whereas the second best is underlined.

	ACDC				FLARE				Friedman	Final
	DSC	HD	ECE	TACE	DSC	HD	ECE	TACE	Rank	Rank
FL [14] ($\gamma = 3$)	0.620	7.30	0.153	0.224	0.834	6.65	0.053	0.145	7.88	8
ECP [28] ($\lambda = 0.1$)	0.782	4.44	0.130	0.151	0.860	<u>5.30</u>	0.037	0.134	5.38	7
LS [32] ($\alpha = 0.1$)	0.809	3.30	0.083	0.093	0.860	5.33	0.055	0.050	4.88	4
SVLS IPMI'21 [10]	0.824	2.81	0.091	0.138	0.857	5.72	0.039	0.144	5.25	5
MbLS CVPR'22 [15]	0.827	2.99	0.103	0.081	0.836	5.75	0.046	0.041	5.25	5
NACL MICCAI'23 [22]	<u>0.854</u>	2.93	0.068	0.073	<u>0.868</u>	**5.12**	0.033	**0.031**	2.25	2
BWCR MICCAI'23 [11]	0.841	<u>2.69</u>	**0.051**	0.075	0.848	5.39	**0.029**	0.059	3.13	3
CRaC (Ours)	**0.877**	**1.72**	<u>0.057</u>	**0.058**	**0.876**	5.52	**0.029**	<u>0.033</u>	1.75	1

Table 2: **Quantitative performance.** Discriminative (DSC ↑, HD ↓) and calibration (ECE ↓, TACE ↓) using nnU-Net [9] as segmentation backbone. The best method is highlighted in bold, whereas the second best is underlined.

	ACDC				FLARE				Friedman	Final
	DSC	HD	ECE	TACE	DSC	HD	ECE	TACE	Rank	Rank
FL [14] ($\gamma = 3$)	0.874	1.60	0.134	0.136	0.893	3.93	0.039	0.061	6.00	6
ECP [28] ($\lambda = 0.1$)	0.889	<u>1.44</u>	0.067	0.112	0.873	5.85	0.046	0.131	6.00	6
LS [32] ($\alpha = 0.1$)	**0.891**	**1.35**	0.067	0.066	0.891	3.61	0.062	0.047	4.00	4
SVLS IPMI'21 [10]	0.883	1.69	0.059	0.111	0.894	4.02	0.026	0.115	5.13	5
MbLS CVPR'22 [15]	0.886	1.46	0.057	0.052	0.891	3.65	0.031	0.031	3.50	3
NACL MICCAI'23 [22]	0.884	1.52	<u>0.056</u>	0.059	**0.896**	<u>3.34</u>	**0.025**	**0.026**	2.50	2
BWCR MICCAI'23 [11]	0.864	1.82	0.063	0.079	0.868	4.47	0.041	0.099	6.63	8
CRaC (Ours)	**0.891**	1.48	**0.052**	**0.051**	<u>0.895</u>	**3.24**	0.029	<u>0.029</u>	1.88	1

first 50 epochs, and reduced to 10^{-4} afterwards. Following [22], the models are trained on 2D slices, and the evaluation is performed over 3D volumes.

Evaluation. **Segmentation:** we employ common segmentation metrics in the medical domain, such as the DICE coefficient (DSC) and the 95% Hausdorff Distance (HD). **Calibration:** following recent works [22,24] we resort to the expected calibration error (ECE) [25] on foreground classes, as in [10], and Thresholded Adaptive Calibration Error (TACE) (threshold of 10^{-3}) [26]. We further compute the Friedman rank [6], to fairly compare the performance of different algorithms in various settings. More details are given in the Appendix.

Comparison to state-of-the-art calibration approaches. In Table 1 and 2, we present the quantitative results of our approach compared to a list of relevant state-of-the-art calibration approaches, when using U-Net and nnU-Net as segmentation backbones, respectively. In terms of **segmentation performance**, our proposed CRaC brings very competitive performance, typically ranking as best, or second best approach, regardless of the segmentation backbone employed. Regarding **calibration**, the trend observed is similar, with CRaC providing well-calibrated models, either improving or at par with state-of-the-art for calibration. Furthermore, as it is common in evaluating many methods in multiple settings, we assess the **overall performance** with a multi-criteria analysis, the Friedman Rank. The results from this metric, which are reported at the right-most columns of both Tables 1 and 2, show that CRaC ranks at the first position, outperforming existing methods when a trade-off between calibration and segmentation performance is considered. Furthermore, the first rank position is maintained even when employing a more powerful backbone, i.e., nnU-Net, consistently delivering the better segmentation-calibration compromise.

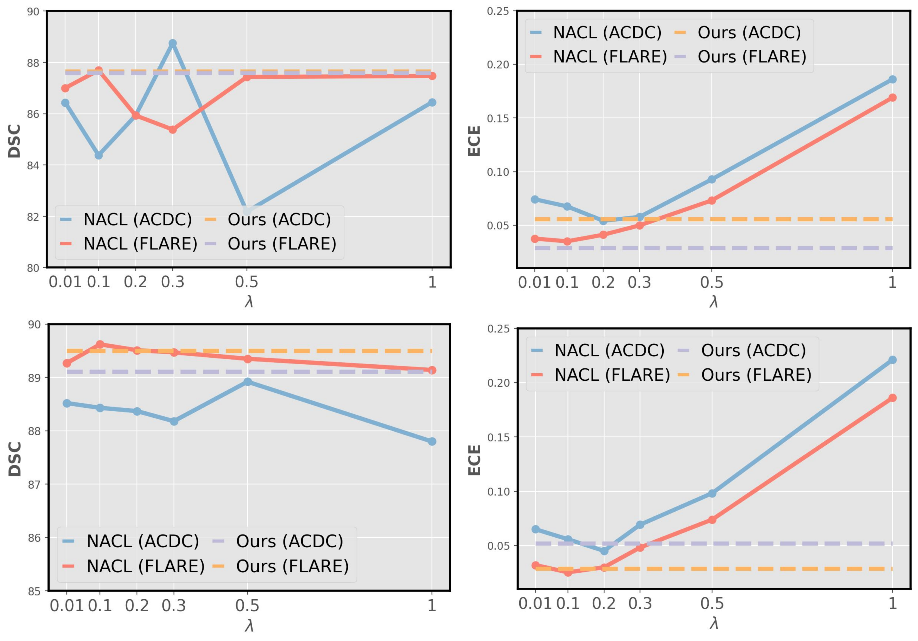

Fig. 1: **Instability of NACL fine-tuning**. Discriminative (*left*) *vs.* calibration performance (*right*) as a function of λ in NACL [22], for both U-Net (*top*) and nnU-Net (*bottom*).

Benefits compared to NACL. In this section we compare the sensitivity of NACL [22] to the choice of its λ value in Eq. 1, as our approach improves NACL by incorporating a mechanism to learn and adapt the class and region-wise penalty terms λ_{kr} in Eq. 3. We found that, despite performing at par in some settings, the performance of NACL significantly varies with the value of its penalty weight which, in addition, is dataset-dependent (Figure 1).

For example, the left-top plot (U-Net) demonstrates that while setting $\lambda = 0.3$ in NACL yields the best discriminative performance in ACDC, it is substantially deteriorated in the FLARE dataset. Furthermore, the λ value that optimizes the discriminative performance (*left* plots) may not be the same that minimizes the miscalibration issue (*right* plots). We also note that these observations hold for both U-Net (*top*) and nnU-Net (*bottom*). Thus, while one may argue that by fine-tuning λ in NACL can lead to improvements over CRaC (and only in certain settings), we advocate that performing a validation search in a dataset-basis is impractical for real-world problems, making of our approach an appealing solution.

4 Conclusion.

We presented a novel approach to calibrate segmentation networks, which accounts for the inherent difficulties of different classes and regions. To address this issue, our method integrates class and region-adaptive constraints, whose penalty weights are learned during training via an Augmented Lagrangian method. Results demonstrate that our approach outperforms existing approaches, becoming an excellent alternative to deliver high-performing and robust models.

Acknowledgments. This work is supported by the National Science and Engineering Research Council of Canada (NSERC), via its Discovery Grant program. We also thank Calcul Quebec and Compute Canada.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article

References

1. Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.A., Cetin, I., Lekadir, K., Camara, O., Ballester, M.A.G., et al.: Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved? IEEE TMI **37**(11), 2514–2525 (2018)
2. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Optimization and Neural Computation Series, Athena Scientific, 1st edn. (1996)
3. Birgin, E.G., Castillo, R.A., Martínez, J.M.: Numerical comparison of augmented lagrangian algorithms for nonconvex problems. Computational Optimization and Applications **31**(1), 31–55 (2005)
4. Cheng, J., Vasconcelos, N.: Calibrating deep neural networks by pairwise constraints. In: CVPR. pp. 13709–13718 (2022)
5. Ding, Z., Han, X., Liu, P., Niethammer, M.: Local temperature scaling for probability calibration. In: ICCV (2021)
6. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the american statistical association **32**(200), 675–701 (1937)
7. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: ICML (2017)
8. Gupta, K., Rahimi, A., Ajanthan, T., Mensink, T., Sminchisescu, C., Hartley, R.: Calibration of neural networks using splines. In: ICLR (2020)
9. Isensee, F., et al.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18** (2020)
10. Islam, M., Glocker, B.: Spatially varying label smoothing: Capturing uncertainty from expert annotations. In: IPMI. pp. 677–688. Springer (2021)

11. Karani, N., Dey, N., Golland, P.: Boundary-weighted logit consistency improves calibration of segmentation networks. In: MICCAI. pp. 367–377 (2023)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. ICLR (2015)
13. Larrazabal, A.J., Martínez, C., Dolz, J., Ferrante, E.: Maximum entropy on erroneous predictions: Improving model calibration for medical image segmentation. In: MICCAI. pp. 273–283 (2023)
14. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: CVPR (2017)
15. Liu, B., Ben Ayed, I., Galdran, A., Dolz, J.: The devil is in the margin: Margin-based label smoothing for network calibration. In: CVPR (2022)
16. Liu, B., Rony, J., Galdran, A., Dolz, J., Ben Ayed, I.: Class adaptive network calibration. In: CVPR. pp. 16070–16079 (2023)
17. Ma, J., Zhang, Y., Gu, S., Zhu, C., Ge, C., Zhang, Y., An, X., Wang, C., Wang, Q., Liu, X., Cao, S., Zhang, Q., Liu, S., Wang, Y., Li, Y., He, J., Yang, X.: Abdomenct-1K: Is abdominal organ segmentation a solved problem? IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
18. Márquez-Neila, P., Salzmann, M., Fua, P.: Imposing hard constraints on deep networks: Promises and limitations. arXiv preprint arXiv:1706.02025 (2017)
19. Minderer, et al.: Revisiting the calibration of modern neural networks. In: NeurIPS (2021)
20. Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P.H., Dokania, P.K.: Calibrating deep neural networks using focal loss. In: NeurIPS (2020)
21. Müller, R., Kornblith, S., Hinton, G.: When does label smoothing help? In: NeurIPS (2019)
22. Murugesan, B., Adiga Vasudeva, S., Liu, B., Lombaert, H., Ben Ayed, I., Dolz, J.: Trust your neighbours: Penalty-based constraints for model calibration. In: MICCAI. pp. 572–581 (2023)
23. Murugesan, B., Liu, B., Galdran, A., Ayed, I.B., Dolz, J.: Calibrating segmentation networks with margin-based label smoothing. Medical Image Analysis **87**, 102826 (2023)
24. Murugesan, B., Vasudeva, S.A., Liu, B., Lombaert, H., Ayed, I.B., Dolz, J.: Neighbor-aware calibration of segmentation networks with penalty-based constraints. arXiv preprint arXiv:2401.14487 (2024)
25. Naeini, M.P., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
26. Nixon, J., Dusenberry, M.W., Zhang, L., Jerfel, G., Tran, D.: Measuring calibration in deep learning. In: CVPR workshops. vol. 2 (2019)
27. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York, NY, USA, 2nd edn. (2006)
28. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. In: ICLR (2017)
29. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241 (2015)
30. Rony, J., Granger, E., Pedersoli, M., Ben Ayed, I.: Augmented lagrangian adversarial attacks. In: CVPR. pp. 7738–7747 (2021)
31. Silva-Rodriguez, J., Hajimiri, S., Ayed, I.B., Dolz, J.: A closer look at the few-shot adaptation of large vision-language models. arXiv preprint arXiv:2312.12730 (2023)
32. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)

33. Thulasidasan, S., Chennupati, G., Bilmes, J., Bhattacharya, T., Michalak, S.: On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In: NeurIPS (2019)
34. Tomani, C., Gruber, S., Erdem, M.E., Cremers, D., Buettner, F.: Post-hoc uncertainty calibration for domain drift scenarios. In: CVPR (2021)
35. Zhang, J., Kailkhura, B., Han, T.: Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In: ICML (2020)