Local Temperature Scaling for Probability Calibration

Zhipeng Ding Xu Han Peirong Liu Marc Niethammer
University of North Carolina at Chapel Hill, Chapel Hill, USA
{zp-ding, xhs400, peirong, mn}@cs.unc.edu

Abstract

For semantic segmentation, label probabilities are often uncalibrated as they are typically only the by-product of a segmentation task. Intersection over Union (IoU) and Dice score are often used as criteria for segmentation success, while metrics related to label probabilities are not often explored. However, probability calibration approaches have been studied, which match probability outputs with experimentally observed errors. These approaches mainly focus on classification tasks, but not on semantic segmentation. Thus, we propose a learning-based calibration method that focuses on multi-label semantic segmentation. Specifically, we adopt a convolutional neural network to predict local temperature values for probability calibration. One advantage of our approach is that it does not change prediction accuracy, hence allowing for calibration as a post-processing step. Experiments on the COCO, CamVid, and LPBA40 datasets demonstrate improved calibration performance for a range of different metrics. We also demonstrate the good performance of our method for multi-atlas brain segmentation from magnetic resonance images.

1. Introduction

With the development of deep convolutional neural networks (CNNs), the accuracy of semantic segmentation has improved dramatically [9, 43]. However, ideally semantic segmentation networks should not only be accurate, but should also indicate when they are likely incorrect. For example, an autonomous driving system might use deep convolutional neural networks to analyze a real-time scene from a camera [5], the associated semantic segmentation of street scenes should provide accurate detections of pedestrians and other vehicles, and the system should recognize when such predictions are unreliable. Another example is the segmentation of brain tumors with a CNN [22]. If the segmentation network can not confidently segment critical regions of the brain, then a medical expert should decide or be alerted to such doubtful regions. Thus, it is important for semantic segmentation networks to generate both accurate label predictions *and* accurate confidence measures.

However, due to overfitting, CNNs for semantic segmentation tend to be overconfident about predicted labels [17, 20, 29, 41]. Approaches for joint prediction and calibration exist [36, 44, 48, 52]. However, they require changing the learning task and typically strive for calibration, but do not guarantee it. An alternative approach is to calibrate the resulting probabilities of a model via *post-processing* so that they better reflect the true probabilities of being correct. This is the kind of approach we consider here as it easily applies to pre-trained networks and can even benfit joint prediction/calibration approaches. Probability calibration, first studied for classification [58], generally addresses this problem via a hold-out validation dataset.

Existing calibration approaches still have several limitations: (1) Most of the probability calibration approaches are designed for classification, thus are not guaranteed to work well for semantic segmentation (where it is also more challenging to annotate on a pixel/voxel level); (2) While there is limited work discussing probability calibration for semantic segmentation, this work either only applies to specific types of models (e.g., Bayesian neural networks [29]) or only implicitly improves calibration performance (e.g., via model ensembling [47] or multi-task learning [31]); (3) Most methods are designed to work for *binary* classifications and approach multi-class problems by a decomposition into k one-vs-rest binary calibrations (where k denotes the number of classes). However, such a decomposition does not guarantee overall calibration (only for the individual subproblems before normalization) and the classification accuracy of the trained model may change after calibration as the probability order of labels may change.

Our goal is to develop a *post-processing* calibration method for multi-label semantic segmentation, which retains label probability order and, therefore, a model's segmentation accuracy. Our work is inspired by temperature scaling (TS) [20] for classification probability calibration. As TS determines only *one* global scaling constant, it cannot capture spatial miscalibration changes in images. We therefore (1) extend TS to multi-label semantic segmentation and (2) make it adaptive to local image changes.

Our contributions are: (1) *Spatially localized probability calibration:* We propose a learning-based local TS method that predicts a separate temperature scale for each pixel/voxel. (2) *Completely separated accuracy-preserving post-processing:* Our approach is completely separated from the segmentation task, leaving the prediction accuracy unchanged. (3) *Theoretical justification:* We provide a theoretical analysis for the effectiveness of our approach. (4) *Comprehensive analysis:* We provide definitions and evaluation metrics for probability calibration for semantic segmentation and validate our approach both qualitatively and quantitatively. (5) *Practical application:* We successfully apply our calibrated probabilities for multi-atlas segmentation label fusion in the field of medical image analysis.

2. Related Work

A variety of calibration approaches have been proposed, but none addresses our target setting.

Bin-based Approaches. Non-parametric histogram binning [67] uses the average number of positive-class samples in each bin as the calibrated probability. Isotonic regression [68] extends this approach by jointly optimizing bin boundaries and bin predictions; it is one of the most popular non-parametric calibration methods. ENIR [55] further extends isotonic regression by relaxing the monotonicity assumption of isotonic regression. These bin-based methods do not consider correlations among neighboring pixels/voxels in semantic segmentation, while our proposed method captures correlations via convolutional filters.

Temperature Scaling Approaches. Platt scaling [58] uses logistic regression for probability calibration. Matrix scaling [20], vector scaling [20], and temperature scaling [25, 20] all generalize Platt scaling to multi-class calibration, among which temperature scaling is both effective and the simplest. ATS [51] extends temperature scaling by using the conditional distribution on each class to address the calibration challenge on small validation datasets, for noisy labels, and highly accurate networks. BTS [30] extends temperature scaling to a bin-wise setting and also uses data augmentation inside each bin to improve the calibration performance. However, unlike our approach (which extends temperature scaling) none of these approaches considers spatial variations for probability calibration.

Bayesian Approaches. BBQ [54] extends binning via Bayesian averaging of the probabilities produced by all possible binning schemes. Bayes-Iso [1] extends isotonic regression by using Bayesian isotonic calibration to allow for more flexibility in the monotonic fitting and smoothness. Jena et al. [29] proposed to use a utility function focusing on the intermediate-layers of a Bayesian deep neural network to calibrate probabilities for image segmentation. Maronas et al. [46] proposed decoupled Bayesian neural networks to calibrate classification probabilities. Bin-based Bayesian methods do not consider pixel/voxel correlations. Bayesian neural networks can capture spatial correlations, but require a Bayesian formulation in the first place. Furthermore, while Bayesian uncertainty quantification [32] helps probability calibration, it may also not achieve it (Appx. A). Instead, our approach considers pixel/voxel correlations and can be used as a post-processing approach for any semantic segmentation method which generates probability outputs.

Other Approaches. Mehrtash et al. [47] found that model ensembling improves confidence calibration for medical image segmentation. A similar conclusion was also found in [38, 69], where an ensemble is used to produce good predictive uncertainty estimates. Karimi et al. [31] showed that multi-task learning can yield better-calibrated predictions than dedicated models trained separately. Note that ensembling or multi-task learning does not directly address probability calibration, instead they provide insights on how to obtain a better calibrated segmentation model. Leathart et al. [39] improved the calibration of classification tasks by building a decision tree over input tabular data, where the leaf nodes correspond to different calibration models. Further, beta calibration [35] extends logistic calibration to overcome the situation where per-class score distributions are heavily skewed. Dirichlet calibration [34] uses the Dirichlet distribution to generalize beta calibration to multi-class problems. Rahimi et al. [59] proposed to use neural network based intra order-preserving functions for calibration. These methods are also not directly designed for probability calibration of semantic segmentation, but focus on classification. Learning algorithms [36, 44, 48, 52] that jointly consider prediction and calibration also exist. Although they can help mitigate miscalibrations, they typically cannot entirely remove it. In fact, they can also benefit from our post-processing approach (§4.2).

3. Methodology

3.1. Problem Statement

Our goal is the calibration of the predicted probabilities of deep semantic segmentation CNNs. Assume there is a pre-trained neural network $\mathcal{F}$, with an image I as the input, which outputs a vector of logits at each location x. Each logit corresponds to a label, and the logit value reflects the label confidence. The predicted label is the one with the largest logit value; the corresponding confidence (probability of correctness) for each pixel/voxel is usually obtained via softmax of the logits. Specifically, the predicted confidence map and the corresponding segmentation map are

$$\hat{P}(x) = \max_{l \in L} \sigma_{SM}(\mathbf{z}(x))^{(l)} = \max_{l \in L} \frac{\exp(\mathbf{z}(x)^{(l)})}{\sum_{j \in L} \exp(\mathbf{z}(x)^{(j)})},$$
$$\hat{S}(x) = \arg\max_{l \in L} \mathbf{z}(x)^{(l)}, \tag{3.1}$$

where σ_{SM} is the softmax function, x denotes position, L is the set of all labels, l is the label index and $\mathbf{z}(x)^{(l)} = z_l(x)$ is the logit that corresponds to label l at location x.

The goal of probability calibration is to ensure that the confidence map $\hat{P}$ represents a true probability. For example, given a 10×10 image, with label confidence of 0.7 for each pixel, we would expect that 70 pixels should be correctly segmented. This can be formalized as follows:

Definition 1. *A semantic segmentation is perfectly calibrated in region Ω if*

$$\mathbb{P}(\hat{S}(x) = S(x)|\hat{P}(x) = p) = p, \forall p \in [0, 1], x \in \Omega \quad (3.2)$$

where $S(x)$ and $\hat{S}(x)$ are the true and predicted segmentations at location x, respectively, $\hat{P}(x)$ is the confidence of the prediction $\hat{S}(x)$, and $\mathbb{P}$ is the probability measure.

In short, if the observed probability is the true probability, then the semantic segmentation model is well-calibrated. As it is difficult to work directly with this definition to assess miscalibration, we extend several visual and quantitative metrics [11, 53, 54, 56, 57], which have previously been proposed in the context of classification.

3.2. Calibration Setup

Assume the data split for a semantic segmentation network $\mathcal{F}$ is D_{train} / D_{val} / D_{test}, i.e. $\mathcal{F}$ is trained on the D_{train} dataset, validated on the D_{val} dataset to choose the best model, and finally tested on the D_{test} dataset. Note that D_{train}, D_{val}, and D_{test} are disjoint datasets. Miscalibration can be observed when evaluating $\mathcal{F}$ on D_{test} for probability-related measures. Our goal is to calibrate the probability output of $\mathcal{F}$ on D_{test}. To this end, we train a calibration model $\mathcal{C}$ on the hold-out validation dataset D_{val} via cross entropy loss, to obtain a better calibrated probability output of $\mathcal{F}$ on D_{test}.

3.3. TS for Probability Calibration

Temperature scaling [20] has been proposed as a simple extension of Platt scaling [58] for post-hoc probability calibration for multi-class classifications. Specifically, temperature scaling estimates a single scalar parameter $T \in \mathbb{R}^+$, i.e., the temperature, to calibrate probabilities: $\hat{q} = \max_{l \in L} \sigma_{SM}(\mathbf{z}/T)^{(l)}$, where $\hat{q}$ is the calibrated probability.

We can directly extend temperature scaling to semantic segmentation by estimating *one* global parameter $T \in \mathbb{R}^+$ for all pixels/voxels of all images: $\hat{\mathbb{Q}}_i(x, T) = \max_{l \in L} \sigma_{SM}(\mathbf{z}_i(x)/T)^{(l)}$, where $\hat{\mathbb{Q}}_i$ is the calibrated probability map for the i-th image. As in [20], we obtain this optimal value for T by minimizing the following negative log-likelihood (NLL) w.r.t. a hold-out validation dataset:

$$T^* = \arg\min_T \left(-\sum_{i=1}^{n} \sum_{x \in \Omega} \log \left(\sigma_{SM}(\mathbf{z}_i(x)/T)^{(S_i(x))} \right) \right) \quad s.t. \quad T > 0, \quad (3.3)$$

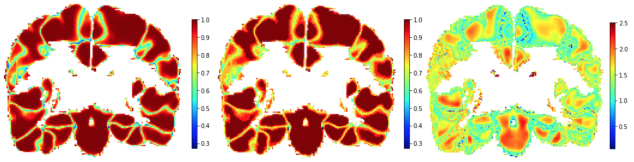

Figure 1: Left: Predicted probabilities (confidence) by a U-Net in §4.3. Middle: Average accuracy of each bin for 10 bins of reliability diagram with an equal bin width indicating different probability ranges that need to be optimized for different locations. Right: Temperature value map obtained via optimization, revealing different optimal localized TS values at different locations.

where Ω denotes the image space and n the number of training images. However, temperature scaling in this way assumes that each image has the same distribution (i.e., the same temperature, T, for all images), which is unrealistic. We therefore propose to relax this assumption as follows:

Definition 2. *Image-based temperature scaling (IBTS):*

$$\hat{\mathbb{Q}}_i(x, T_i) = \max_{l \in L} \sigma_{SM}(\mathbf{z}_i(x)/T_i)^{(l)}, \quad (3.4)$$

where $T_i \in \mathbb{R}^+$ is image-dependent.

While this at first seems like a minor change to the standard temperature scaling approach, it is important to note that moving to an image-based temperature value, T_i requires us to *learn* a regressor which predicts this temperature value for each image, I. Therefore, we use a CNN [19] to learn a mapping from $(\mathbf{z}_i, I_i)$ to T_i. Suppose the network is $\mathscr{F}$, then the optimization is

$$\theta^* = \arg\min_\theta -\sum_{i=1}^{n} \sum_{x \in \Omega} \log \left(\sigma_{SM}\left(\frac{\mathbf{z}_i(x)}{\mathscr{F}(\theta, \mathbf{z}_i, I_i)} \right)^{(S_i(x))} \right) \quad s.t. \quad \mathscr{F}(\theta, \mathbf{z}_i, I_i) > 0, \quad (3.5)$$

where θ are the parameters of the network $\mathscr{F}$. The calibrated probability can be obtained by substituting $T_i^* = \mathscr{F}(\theta^*, \mathbf{z}_i, I_i)$ in Eq. (3.4).

3.4. Local TS for Probability Calibration

Probabilities predicted by a deep CNN vary by location. Fig. 1 illustrates that object interiors can usually be accurately predicted while predictions on boundary or near-boundary locations are more ambiguous. Thus the optimal temperature value may vary across locations. However, using a global parameter, T, or an image-based parameter, T_i, cannot account for such spatial variations. That this is a practical concern is illustrated in the uncalibrated reliability diagrams of Fig. 2 which shows that the confidence-vs-accuracy relation may indeed vary across an image. Hence, spatial variations should be considered for semantic segmentation. Therefore, we propose the following local temperature scaling (LTS) approach.

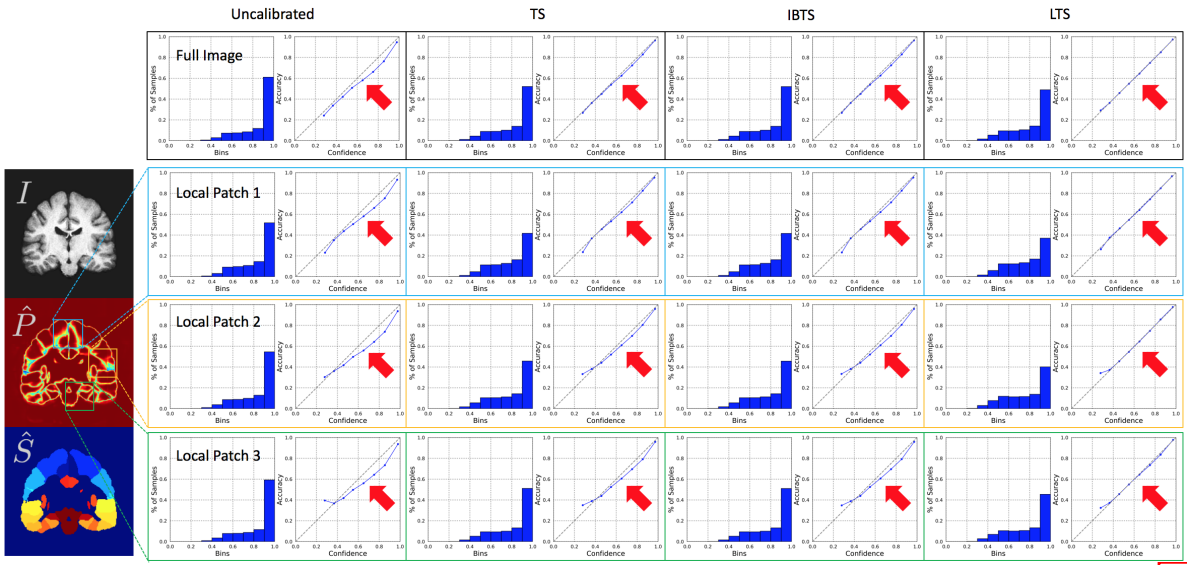

Figure 2: An example of global and local reliability diagrams for different methods for a U-Net segmentation experiment (§4.3). I is the image, $\hat{P}$ is the predicted uncalibrated probability, and $\hat{S}$ is the predicted segmentation. Figures are displayed in couples, where the left figure is the probability distribution of pixels/voxels while the right figure is the reliability diagram (See Appx. F for definitions). The top row shows the global reliability diagrams for different methods for the entire image. The three rows underneath correspond to local reliability diagrams for the different methods for different local patches. Note that TS and IBTS can calibrate probabilities well across the entire image. Visually, they are only slightly worse than LTS. However, when it comes to local patches, LTS can still successfully calibrate probabilities while TS and IBTS can not. In general, LTS improves local probability calibrations. More results are in Appx. D.

Definition 3. *Local temperature scaling (LTS):*

$$\hat{\mathbb{Q}}_i(x, T_i(x)) = \max_{l \in L} \sigma_{SM}(\mathbf{z}_i(x)/T_i(x))^{(l)}, \qquad (3.6)$$

where $T_i(x) \in \mathbb{R}^+$ *is image* and *location dependent.*

For $T_i(x) = 1$, no calibration occurs as the logits $\mathbf{z}_i(x)$ do not change. For $T_i(x) > 1$, confidence will be reduced, which helps counteract overconfident predictions. As $T_i(x) \to \infty$, the calibrated probabilities will approach $1/|L|$, which represents maximum uncertainty. For $T_i(x) < 1$, prediction confidence will be increased. This will be helpful to counteract underconfident predictions. Lastly, as $T_i(x) \to 0$, the calibrated probabilities will become binary ($\in \{0, 1\}$), which represents minimum uncertainty. As $T_i(x)$ is positive, such a local scaling does not change the ordering of the probabilities over the different classes. Hence, the segmentation accuracy remains unchanged.

Another network $\mathscr{H}$, with parameter α, can be used to learn this local mapping from $(\mathbf{z}_i, I_i)$ to $T_i(x)$. The optimization follows Eq. (3.5), with $\mathscr{F}(\theta, \mathbf{z}_i, I_i)$ replaced by $\mathscr{H}(\alpha, \mathbf{z}_i, I_i, x)$, where x indicates the spatial locations. Finally, we obtain $T_i(x)^* = \mathscr{H}(\alpha^*, \mathbf{z}_i, I_i, x)$.

Fig. 3 illustrates our high-level design for probability calibration. The input is a logit map $\mathbf{z}$, usually obtained by a segmentation network (Seg). Together with the image I, it is then passed to an optimization unit or a prediction unit to generate the temperature map. These temperature values are used to calibrate the logit map. The calibrated probabilities

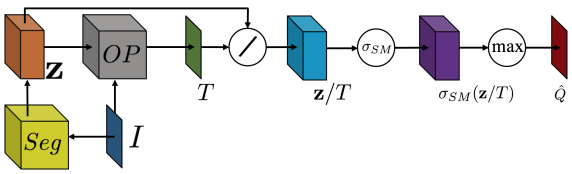

Figure 3: Architecture for probability calibration via (local) temperature scaling. The output logit map of a pre-trained semantic segmentation network (Seg) is locally scaled to produces the calibrated probabilities. OP denotes optimization or prediction via a deep convolutional network to obtain the (local) temperature values. Details of this OP unit can be found in Appx. B.

are, in turn, obtained via a softmax on the calibrated logits. Class labels do not change under this process and can still be obtained by determining the class with the largest predicted probability. Appx. B details the implementation. Training details are described in Appx. C.

3.5. Theoretical Justification

Why does miscalibration happen? One usually uses the loss corresponding to the negative log-likelihood (NLL) of the multinomial distribution [3, 15] (i.e., the multi-class cross-entropy loss) to train semantic segmentation networks because minimizing it will minimize the Kullback-Leibler (KL) divergence between the ground-truth probability distribution and the predicted probability distribution. The minimum loss is achieved if and only if the predicted probability distribution recovers the ground-truth probability dis-

tribution [3, 15]. For semantic segmentation, the NLL loss is minimized when $\hat{P}(x) = 1$ and $\hat{S}(x) = S(x)$, for all x. The segmentation error is minimized when $\mathbf{z}(x)^{(S(x))} > \mathbf{z}(x)^{(l)}$ for all $l \in L$ and $l \neq S(x)$. This indicates that even if the segmentation error is minimized to zero, the NLL loss may still be positive and the optimization will consequently try to continue reducing it to zero by pushing $\hat{P}(x)$ to one for $\hat{S}(x) = S(x)$. This explains how overconfidence occurs in the context of semantic segmentation. Note that this overconfidence also results in low-entropy distributions.

How to eliminate miscalibration? As indicated in [52] encouraging the predicted distribution to have higher entropy can help avoid overconfident predictions for deep CNNs, and can thereby improve calibration. Thus, to calibrate an overconfident semantic segmentation network, we need to simultaneously minimize the NLL loss w.r.t. the to-be-learned calibration parameters while assuring that the corresponding entropy of the calibrated probabilities stays sufficiently large to probabilistically describe empirically observable segmentation errors. Note that we minimize the NLL loss for the same reason as for segmentation (above): because the goal is to recover the true probability distribution. The difference is that for segmentation we optimize w.r.t. the segmentation network parameters while for calibration we optimize w.r.t. the calibration model parameters.

Why do we use (local) TS to calibrate probabilities? Overconfident networks usually exhibit the phenomenon that the entropy of the output probabilities is much lower than the cross entropy on the testing dataset as shown in [20, 52]. Thus, we define overconfidence as entropy being lower than the cross entropy of probabilities (Appx. E; and similarly for underconfidence). Specifically, we show the following theorem in Appx. E.

Theorem 4. *When the to-be-calibrated segmentation network is overconfident, minimizing NLL w.r.t. TS, IBTS, and LTS results in solutions that are also the solutions of maximizing entropy of the calibrated probability w.r.t. TS, IBTS and LTS under the condition of overconfidence.*

For example, for TS, the above theorem can be mathematically expressed as follows,

$$\arg\min_{T} -\sum_{i=1}^{n}\sum_{x\in\Omega}\log\left(\sigma_{SM}\left(\mathbf{z}_i(x)/T\right)^{(S_i(x))}\right)$$

$$\Updownarrow$$

$$\arg\max_{T} -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\frac{\mathbf{z}_i(x)}{T}\right)^{(l)}\log\left(\sigma_{SM}\left(\frac{\mathbf{z}_i(x)}{T}\right)^{(l)}\right)$$

$$s.t.\ \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\frac{\mathbf{z}_i(x)}{T}\right)^{(l)} \geq \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}$$

where $T > 0$. Hence, our three different variants for probability calibration via temperature scaling (TS, IBTS, LTS) will counteract the tendency of entropy minimization caused by the NLL loss discussed above. Training the segmentation network via the NLL loss followed by post-hoc probability calibration via temperature scaling is an effective approach to obtain high segmentation accuracy while avoiding overconfidence of the resulting label probabilities. §4.1-§4.4 show experiments to support this claim.

4. Experiments

We show the performance and behavior of our proposed TS approaches for semantic segmentation on the COCO dataset (§4.1), CamVid dataset (§4.2) and LPBA40 dataset (a dataset of magnetic resonance (MR) images of the human brain) (§4.3). We further show how our probability calibration may influence downstream tasks, by exploring it in the context of multi-atlas segmentation on LPBA40 (§4.4).

Evaluation Metrics. To assess the performance of probability calibration, we use five metrics, which were originally designed for classification, for semantic segmentation. Specifically, they are the reliability diagram [11, 53, 56], expected calibration error [54] (ECE), maximum calibration error [54] (MCE), static calibration error [57] (SCE), and adaptive calibration error [57] (ACE). To make the above metrics applicable to semantic segmentation, we consider the predicted probabilities for each pixel/voxel as separate samples. We use 10 equally-sized (probability or sample size) bins to compute all these metrics. In §4.4, we additionally use average surface distance (ASD), surface Dice (SD), the 95-th percentile of the maximum symmetric distance (95MD), and average volume Dice (VD) to measure segmentation performance. Detailed definitions are in Appx. F.

Baseline Methods. To illustrate the effectiveness of our proposed LTS approach (see Eq. (3.6)), we compare it to standard TS and IBTS (see Eq. (3.4)), where we directly assess if local adjustments can be properly predicted and if they are beneficial. While other probability calibration methods exist, as discussed in §2, most are for classification and not for semantic segmentation. This is an important difference. For example, in semantic segmentation, nearby pixels/voxels are correlated with each other, whereas such relations do not apply to classification. Thus, simply considering each pixel/voxel as a classification data point is not appropriate. For completeness, however, we still choose several classic methods (§4.1) to compare against, i.e. isotonic regression (IsoReg) [68], vector scaling (VS) [20], ensemble temperature scaling (ETS) [69], and Dirichlet calibration with off-diagonal regularization (DirODIR) [34]. Furthermore, to illustrate that our method is also beneficial for joint training (§4.2), we show the performance before and after using LTS for models trained with maximum mean calibration loss (MMCE) [36] and focal loss (FL) [52]. All methods are fine-tuned with the best parameters via grid search. Details are in Appx. C.

Evaluation Regions. Since label boundaries are difficult

to segment, these are the regions where most of the relevant miscalibrations are expected to occur (see also Fig. 1). For a refined analysis, we extract boundaries and their nearby regions (i.e., regions up to 2 pixels/voxels away from the boundary). We denote this evaluation region by *Boundary* in all experiments. We also evaluate performance *within* label regions (excluding the background, but including the respective *Boundary* region). We denote this large region as *All*. It is expected that the calibration inside the *Boundary* region will be more challenging (as the prediction is more ambiguous) than the calibration inside the bigger *All* region. Appx. G shows examples of these regions for a 3D brain MR image. Furthermore, to evaluate the local probability calibration performance for an image segmentation, we also randomly select 10 small patches (72×72 for 2D, $72\times72\times72$ for 3D) and compute the same metrics as for the entire image. We report average performance (denoted *Local-Avg*) and the worst case performance (denoted *Local-Max*) across 10 patches. Appx. H shows results for different patch sizes. Note that results in the *All* region reflect the overall calibration performance for an image segmentation; results in the *Boundary* region reflect the most challenging calibration performance for an image segmentation; results in the *Local* region generally reflect whether the calibration method can handle spatial variations.

Downstream MAS setting. Multi-atlas segmentation (MAS) relies on transferring segmentations from a set of atlas images to a target image via deformable registration. The segmentation in the target space is then obtained by a label fusion method, which establishes a consensus among the registered atlas labels. We use the label fusion strategy by Wang et al. [64], which takes advantage of the label probabilities. Hence, better-calibrated probabilities should lead to better fusion accuracy (i.e., segmentation accuracy).

Statistical Considerations. To indicate the success of probability calibration, we use a Mann-Whitney U-test [45] to check for significant differences between the result of LTS and the results for all other baseline methods (UC, TS, IBTS, etc.). We use the Benjamini/Hochberg correction [4] for multiple comparisons with a false discovery rate of 0.05. Results are highlighted in green when LTS performs significantly better than the corresponding method (no color means no statistically significant differences).

Datasets. We use three datasets for our experiments: The Common Object in Context (COCO) [42] dataset, the Cambridge-driving Labeled Video Database (CamVid) [7, 6], and the LONI Probabilistic Brain Atlas (LPBA40) [62] dataset. Detailed descriptions and the training/validation/testing splits are in Appx. C.

4.1. FCN semantic segmentation on COCO

General: We use a Fully-Convolutional Network (FCN) [43] with a ResNet-101 [23] backbone for semantic segmentation on the COCO dataset. Tab. 1 shows our quantitative evaluation results for calibrating such a segmentation model. In the *All* region, TS and IBTS do not improve calibration performance, possibly because the natural images in the COCO dataset are complex and vary significantly in type and shape, yet TS uses a global temperature value for all images. IBTS performs slightly better than TS on average because it uses an image-dependent temperature scaling to capture image variations, though it cannot explain the spatial image variations in the *All* region. Furthermore, we observe that LTS is in general significantly better than classical methods, i.e. IsoReg [68], VS [20], ETS [69] and DirODIR [34]. This is likely because these classical methods treat each pixel/voxel independently without considering their spatial correlations in semantic segmentation.

Boundary: The relatively low segmentation performance of the segmentation network suggests that such spatial variations might matter. Specifically, semantic segmentation results in a mean IOU of 63.7%, indicating how challenging this dataset is. Further, all methods except VS [20] show significant improvements in the *Boundary* region. This indicates that (1) these boundary regions share common miscalibration patterns, which can be captured by most methods, and (2) miscalibration effects are indeed, as expected, more pronounced in these boundary regions.

Local: Different from the *All* region, the *Local* region is based on randomly extracted small patches of an image. Specifically, *Local-Avg* reflects the average performance of local probability calibration while *Local-Max* reflects the calibration performance in the most uncalibrated patch region thus measuring the worst-case calibration result. Results in ECE, SCE and ACE all suggests that LTS can calibrate the entire image region as well as local image regions. Other approaches result in significantly worse calibrations.

MCE: Further, the MCE results illustrate that probability calibration for semantic segmentation is indeed very challenging compared with classification. This is because classification annotation is typically very accurate while per-pixel/voxel annotation of semantic segmentation can be difficult, especially at object boundaries. For example, in the extreme case, if one pixel/voxel is annotated wrong but predicted correct (or vice versa), then the accuracy is 0 while the prediction confidence is nearly 100%. This will result in MCE values close to 100% for bin based evaluation. Usually, these outliers make up only a small portion of all pixels/voxels in an image. Examples for such *outliers* can be observed in Fig. 2 uncalibrated patch 1 and 3 at the lowest confidence point, where the percentage of samples is very small, but the accuracy-confidence difference is notable. Thus, for all experiments, we expect that MCE can be very high compared to the classification probability calibration literature. LTS can improve MCE values, but may still result in large MCE values.

Dataset	Method	ECE(%)↓			MCE(%)↓			SCE(%)↓			ACE(%)↓		
		All	*Boundary*	*Local-Avg* [*Local-Max*]	*All*	*Boundary*	*Local-Avg* [*Local-Max*]	*All*	*Boundary*	*Local-Avg* [*Local-Max*]	*All*	*Boundary*	*Local-Avg* [*Local-Max*]
FCN COCO (1000)	UC	12.44(17.87)	24.41(7.23)	14.48(20.89) [33.14(26.83)]	27.66(22.23)	38.61(7.22)	34.90(23.89) [58.73(19.66)]	20.24(18.75)	24.97(7.07)	20.05(21.67) [39.66(24.30)]	20.19(18.73)	24.46(7.26)	19.86(21.68) [39.16(24.62)]
	IsoReg [68]	12.55(14.22)	16.27(6.62)	15.35(16.81) [29.26(22.36)]	27.58(21.06)	33.36(10.01)	31.76(20.05) [43.24(23.70)]	22.28(15.35)	17.20(6.42)	21.65(17.77) [37.13(19.38)]	22.19(15.35)	16.40(6.77)	21.41(17.82) [36.69(19.69)]
	VS [20]	12.70(17.22)	24.60(6.98)	14.57(20.26) [29.89(17.28)]	38.40(16.92)	38.96(7.45)	41.20(20.23) [50.42(25.40)]	18.05(18.25)	25.00(6.90)	18.13(21.07) [32.31(18.43)]	17.98(18.25)	24.55(7.09)	17.92(21.07) [32.22(18.40)]
	ETS [69]	12.54(14.27)	15.68(6.79)	15.42(16.88) [29.41(22.44)]	27.36(21.01)	**33.27(10.09)**	30.92(20.34) [42.72(24.68)]	22.37(15.42)	16.72(6.58)	21.80(17.83) [37.33(19.41)]	22.29(15.41)	15.82(6.93)	21.57(17.87) [36.85(19.75)]
	DirODIR [34]	11.32(12.61)	14.17(17.73)	15.09(18.99) [26.85(23.36)]	26.66(18.43)	34.04(12.88)	32.54(24.79) [46.07(18.04)]	19.59(13.16)	15.27(7.75)	18.55(19.44) [34.48(23.17)]	19.67(13.15)	15.33(7.47)	18.71(19.34) [34.46(23.18)]
	TS [20]	12.53(14.28)	15.69(6.79)	15.41(16.89) [29.37(22.47)]	27.27(20.95)	33.27(10.17)	**30.91(20.32)** [42.71(24.66)]	22.36(15.42)	16.73(6.59)	21.78(17.85) [37.34(19.42)]	22.28(15.42)	15.83(6.94)	21.56(17.88) [36.85(19.76)]
	IBTS	11.92(13.83)	16.35(7.13)	14.80(16.63) [28.89(21.99)]	26.25(20.26)	33.29(9.96)	31.19(19.97) [43.45(23.27)]	21.68(15.31)	17.31(6.90)	21.06(17.81) [36.62(19.32)]	21.62(15.29)	16.40(7.33)	20.82(17.84) [36.09(19.63)]
	LTS	**10.04(11.54)**	**13.44(6.23)**	**12.26(14.74)** [**24.31(18.63)**]	**26.17(15.67)**	35.18(12.31)	31.66(17.66) [**40.13(20.39)**]	**16.92(13.89)**	**14.53(6.18)**	**16.78(16.38)** [**30.05(17.45)**]	**16.91(13.93)**	**15.16(5.92)**	**16.85(16.45)** [**30.21(17.60)**]
Tiramisu CamVid (233)	UC	7.79(4.94)	22.79(5.76)	9.23(10.63) [25.35(12.80)]	22.64(12.72)	30.42(10.65)	30.33(16.63) [56.15(14.61)]	9.91(5.02)	24.62(5.69)	13.16(11.72) [30.60(12.48)]	9.90(5.01)	24.43(5.75)	13.15(11.73) [30.60(12.46)]
	TS [20]	3.45(3.52)	12.66(5.43)	7.31(7.72) [17.69(11.91)]	16.02(11.09)	23.57(12.88)	27.29(16.23) [37.25(18.98)]	9.42(3.90)	17.85(4.55)	13.50(10.14) [27.72(11.37)]	9.44(3.92)	17.61(4.59)	13.50(10.17) [27.76(11.33)]
	IBTS	3.63(3.65)	12.57(6.07)	7.25(7.67) [17.60(11.91)]	16.01(10.21)	23.24(13.00)	27.04(15.94) [37.61(19.27)]	9.47(3.89)	17.98(4.88)	13.48(10.12) [27.69(11.38)]	9.49(3.91)	17.75(4.92)	13.48(10.16) [27.76(11.33)]
	LTS	3.40(3.59)	11.80(5.20)	**6.89(7.64)** [16.61(11.81)]	**12.44(7.48)**	22.17(9.53)	27.64(16.67) [37.92(20.47)]	8.76(4.05)	17.77(4.26)	12.66(10.04) [26.78(11.22)]	8.73(4.03)	17.32(4.32)	12.61(10.07) [26.76(11.22)]
	MMCE [36]	4.45(4.03)	–	– [–]	18.83(10.82)	–	– [–]	8.59(5.98)	–	– [–]	8.50(5.00)	–	– [–]
	MMCE [36]+**LTS**	4.15(3.54)	–	– [–]	17.98(10.69)	–	– [–]	7.28(3.80)	–	– [–]	7.17(3.84)	–	– [–]
	FL [52]	3.47(3.11)	**8.68(5.45)**	9.01(7.19) [13.84(11.67)]	14.77(13.28)	**17.62(13.53)**	28.37(15.86) [33.33(18.08)]	7.46(3.43)	**14.08(4.49)**	14.09(9.78) [23.60(12.11)]	7.43(3.45)	**13.63(4.57)**	14.06(9.83) [23.62(12.05)]
	FL [52]+**LTS**	**3.13(3.64)**	11.06(5.55)	6.96(8.21) [**12.66(12.87)**]	14.51(11.07)	19.61(9.82)	**26.91(16.06)** [**32.27(19.08)**]	**6.78(4.05)**	15.28(4.76)	**11.85(10.69)** [**22.04(13.05)**]	**6.73(4.05)**	14.76(4.84)	**11.83(10.73)** [**22.10(12.96)**]
U-Net LPBA40 (40)	UC	5.58(1.16)	14.53(1.67)	5.62(0.95) [10.23(2.82)]	10.71(2.10)	19.18(1.71)	11.74(4.55) [19.46(4.75)]	7.34(1.04)	15.01(1.63)	8.24(3.08) [12.98(2.88)]	7.13(1.02)	14.64(1.62)	8.20(3.06) [12.93(2.83)]
	TS [20]	1.43(0.74)	8.74(1.07)	2.24(1.93) [5.66(2.49)]	4.37(3.73)	14.90(1.74)	6.68(4.44) [11.03(5.31)]	6.47(0.91)	10.06(1.10)	7.81(2.54) [11.49(2.53)]	6.30(0.90)	9.46(1.06)	7.77(2.55) [11.49(2.48)]
	IBTS	1.47(0.77)	8.79(1.14)	2.34(1.98) [5.81(2.46)]	4.40(3.65)	14.96(1.75)	6.79(4.36) [10.84(4.60)]	6.46(0.91)	10.10(1.17)	7.80(2.55) [11.51(2.54)]	6.29(0.90)	9.50(1.13)	7.76(2.56) [11.51(2.49)]
	LTS	**0.90(0.51)**	**7.00(1.23)**	**1.90(1.38)** [**3.70(2.45)**]	**3.51(3.42)**	**12.33(1.96)**	**5.80(3.68)** [**9.29(4.73)**]	**6.27(0.93)**	**8.53(1.04)**	**7.60(2.49)** [**10.89(2.61)**]	**6.09(0.92)**	**7.93(1.08)**	**7.56(2.49)** [**10.87(2.58)**]
VoteNet+ LPBA40 (640)	UC	7.26(0.60)	12.78(0.75)	7.25(2.73) [11.16(1.77)]	12.65(0.76)	19.99(1.10)	12.67(3.14) [16.72(1.63)]	7.29(0.59)	12.79(0.75)	7.35(2.67) [11.22(1.78)]	2.30(0.39)	3.52(0.55)	6.25(2.87) [10.23(1.58)]
	TS [20]	5.07(0.59)	9.48(0.77)	5.08(2.48) [8.77(1.74)]	8.44(0.84)	18.69(1.27)	8.54(3.39) [13.14(2.08)]	5.11(0.58)	9.69(0.80)	5.29(2.39) [8.90(1.78)]	2.12(0.37)	3.38(0.52)	4.62(2.44) [8.21(1.59)]
	IBTS	2.77(0.37)	**4.06(0.45)**	3.14(1.09) [3.21(1.13)]	5.57(0.97)	16.90(2.20)	6.57(2.99) [5.26(2.81)]	3.28(0.39)	**4.27(0.55)**	3.96(1.26) [4.27(1.62)]	0.69(0.26)	2.30(0.40)	3.15(1.06) [3.63(1.12)]
	LTS	**0.71(0.33)**	4.18(0.73)	**1.64(0.94)** [**2.43(1.64)**]	**1.46(0.67)**	**11.55(1.68)**	**3.54(2.02)** [**4.52(3.26)**]	**1.24(0.49)**	4.87(0.83)	**2.52(1.26)** [**3.45(1.94)**]	**0.30(0.24)**	**2.14(0.43)**	**1.90(1.00)** [**2.69(1.35)**]

Table 1: Calibration results for 4 different segmentation models on 4 different tasks. Results are reported in mean(std) format. The number of testing samples are listed in parentheses underneath each dataset name. UC denotes the uncalibrated result. ↓ denotes that lower is better. Best results are bolded and green indicates statistically significant differences w.r.t. LTS (FL+LTS for CamVid). Note that due to GPU memory limits, results of MMCE and MMCE+LTS are for downsampled images, thus can not be directly compared with other methods. The goal of including them is to show that LTS can improve MMCE. LTS generally achieves the best performance on almost all metrics in the *All* region, *Boundary* region and *Local* region. Additional results are in Appx. J.

4.2. Tiramisu semantic segmentation on CamVid

General: We use the Tiramisu segmentation model [28] on the CamVid dataset. Tab. 1 shows quantitative results for calibrating this segmentation model. Compared with the results for the COCO dataset, all four metrics are reduced greatly. This is mainly because the images in CamVid only contain 11 class street scenes and the images are relatively consistent for such scenes. Instead, images from the COCO dataset show different objects in different images. See Appx. I for details. Results are consistent with the COCO dataset. Specifically, (1) LTS can calibrate both the *All* region probabilities as well as the local regions inside an image; (2) LTS is, in general, significantly better than TS and IBTS for most comparisons.

Joint Prediction and Calibration: Further, we show that our approach is beneficial for methods that jointly optimize prediction and calibration [36, 52]. MMCE [36] and FL [52] both consider miscalibration when training semantic segmentation networks. Tab. 1 shows that compared to the uncalibrated results, both MMCE and FL work significantly better. Furthermore, with LTS as a post-hoc calibration, calibration performance further consistently improves (except *Boundary* regions for FL). These findings are consistent with the results in [52] where TS is used as a post-hoc calibration method and the authors show that MMCE+TS and FL+TS work consistently better than MMCE and FL. Hence, this favors our LTS as a successful post-hoc calibration method for segmentation.

4.3. U-Net segmentation on LPBA40

General: We use a customized 3D U-Net [9] for the segmentation of the LPBA40 dataset. Tab. 1 shows quantitative results for calibrating this segmentation model. All three methods calibrate the probabilities relatively well in this experiment. This might be because images have been affinely registered to a common atlas space, which reduces the variations of images and may make it easier for TS, IBTS and, LTS to calibrate both in the *All* region and the *Boundary* region. This might also explain the performance differences between the computer vision datasets and the medical imag-

Method	ASD (mm)↓	SD (%)↑	95MD (mm)↓	VD (%)↑		VC(*All*) (%)			VC(*Boundary*) (%)		
				All	*Boundary*	rate	w→c ↑	c→w ↓	rate	w→c ↑	c→w ↓
Best Fusion	0.04(0.01)	99.06(0.23)	0.18(0.08)	98.99(0.19)	97.29(0.45)	20.53(1.13)	94.62(0.93)	0.00(0.00)	35.85(1.06)	94.11(0.90)	0.00(0.00)
Best Calibration	0.27(0.04)	93.51(1.01)	1.69(0.20)	93.71(0.73)	87.70(1.09)	13.96(0.43)	98.88(0.18)	0.00(0.00)	25.93(0.46)	98.68(0.21)	0.00(0.00)
UC	0.99(0.07)	75.89(1.79)	3.82(0.26)	81.19(1.09)	61.01(1.13)	-	-	-	-	-	-
TS	0.99(0.07)	75.85(1.80)	3.83(0.27)	81.21(1.08)	61.01(1.13)	0.45(0.03)	43.20(1.33)	40.16(1.23)	0.73(0.04)	39.34(1.32)	41.37(1.24)
IBTS	1.00(0.07)	75.75(1.82)	3.86(0.27)	81.20(1.08)	60.87(1.13)	1.43(0.12)	41.14(1.56)	43.27(1.35)	2.35(0.17)	36.93(1.45)	45.14(1.30)
LTS	**0.98(0.07)**	**75.96(1.78)**	**3.82(0.26)**	**81.27(1.07)**	**61.15(1.13)**	1.88(0.14)	42.42(1.43)	37.53(1.04)	2.96(0.18)	40.51(1.15)	35.59(1.01)

Table 2: MAS label fusion results based on calibrated probabilities. ↓(↑) indicates that lower(higher) values are better. mm denotes millimeter. UC denotes uncalibrated results. VC denotes voxel annotation changes between the uncalibrated approach to the corresponding method: w→c is from wrong voxel annotation to correct voxel annotation; c→w is from correct voxel annotation to wrong voxel annotation. Rate is calculated based on the number of changes out of the possible number of changes. (Note that many voxel annotations can not change because all atlas annotations give the same label, thus a change in probability would not change the voxel annotation.) LTS generally improves segmentations slightly. After LTS probability calibration, JLF changes more voxels than for TS and IBTS. Further, the difference between the correct conversion and the incorrect conversion is improved over TS and IBTS. This indicates that JLF can produce better segmentations with a better probability calibration and suggests that downstream tasks may in general benefit from better calibration.

ing dataset in Tab. 1. See Appx. I for details. Differences between calibration performance among TS and IBTS are relatively small. However, LTS still performs best with respect to most metrics.

Spatial Variation: Furthermore, when it comes to the *Local* region analysis, LTS consistently works best. Fig. 2 visualizes such difference via reliability diagrams. The red arrows highlight that TS, IBTS and LTS calibrate probabilties for the whole image well but only LTS consistently performs well in the *Local* region. This indicates the superiority of LTS's spatially-variant probability calibration.

4.4. Downstream MAS label fusion on LPBA40

We use a customized VoteNet+ [13] for multi-atlas segmentation on the LPBA40 dataset. In this approach, a network (VoteNet+) is trained to locally predict if a labeled atlas that has been registered to the target image space should be considered trustworthy or not. Label fusion (among the registered atlas images) can then make use of these probabilities to obtain the multi-atlas segmentation results. It is these VoteNet+ probabilities that we seek to calibrate.

Calibration Metrics: Tab. 1 shows our quantitative calibration results. Different from the U-Net experiments in §4.3, we observe bigger differences between the calibration approaches. This might be because the VoteNet+ calibration experiment has sufficient training data (as multi-atlas segmentation performs image registrations from each atlas image to each target image) whereas the experiments in §4.3 are much more data-starved. Besides, as the labeled atlases are registered to the target image space via a flexible non-parametric registration approach, data variance is further reduced in comparison to the affine registrations used as preprocessing in §4.3. Tab. 1 shows that all three methods calibrate probabilities well, and that performance order is consistent with model complexity. I.e., LTS performs better than IBTS, and IBTS performs better than TS. These differences are statistically significant.

Label Fusion with Probability: Tab. 1 only demonstrates that the calibration approaches can improve the calibration of the VoteNet+ output. To obtain the multi-atlas segmentation result, we need to use label fusion. As the joint label fusion (JLF) approach [64] we use for this purpose can make use of the VoteNet+ label probabilities, it is natural to ask if improved calibration results translate to improved segmentations via JLF. Tab. 2 shows that while differences are small, consistent improvements can indeed be observed. Hence, our proposed LTS not only shows good calibration performance on traditional metrics (i.e. ECE, MCE, SCE and ACE), but can also benefit downstream tasks that are sensitive to accurate probabilities. For comparison, we also show two theoretical upper bounds. The *Best Fusion* bound, which is obtained by assigning the correct label to the segmentation result if at least one atlas provides the right label; and the *Best Calibration* bound, which is obtained by assigning a probability of 1 if the prediction by VoteNet+ is correct and $1/|L|$ otherwise, followed by JLF. We observe that there is still a large room to improve probability calibration as the obtained results are far from the two upper bounds.

5. Conclusion and Future Work

We introduced LTS, a general temperature scaling method that allows for spatially-varying probability calibration for multi-label semantic segmentation. Experiments on the COCO, CamVid and LPBA40 datasets show that LTS outperforms probability calibration approaches which cannot account for spatially-varying miscalibration. LTS not only works for standard segmentation models but can also benefit models that aim to jointly optimize prediction and calibration. Further, using a multi-atlas brain segmentation experiment we demonstrated that downstream tasks may benefit from improved probability calibration. Future work could focus on further calibration improvements. For example, LTS could be easily extended to a bin-wise setting as in [30] or use distributions conditioned on classes as in [51].

Acknowledgements. This work was supported by NIAMS 1R01-AR072013, NIMH 2R42MH118845, and NSF EECS-1711776; it expresses the views of the authors, not of NIH/NSF. The authors have no conflicts of interest.

References

[1] Mari-Liis Allikivi and Meelis Kull. Non-parametric Bayesian isotonic calibration: Fighting over-confidence in binary classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 103–120. Springer, 2019.

[2] Xabier Artaechevarria, Arrate Munoz-Barrutia, and Carlos Ortiz-de Solórzano. Combination strategies in multi-atlas image segmentation: application to brain MR data. *IEEE transactions on medical imaging*, 28(8):1266–1277, 2009.

[3] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press, 2017.

[4] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[6] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[7] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pages 44–57. Springer, 2008.

[8] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[9] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[10] Pierrick Coupé, José V Manjón, Vladimir Fonov, Jens Pruessner, Montserrat Robles, and D Louis Collins. Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation. *NeuroImage*, 54(2):940–954, 2011.

[11] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[12] Zhipeng Ding, Xu Han, and Marc Niethammer. Votenet: A deep learning label fusion method for multi-atlas segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 202–210. Springer, 2019.

[13] Zhipeng Ding, Xu Han, and Marc Niethammer. Votenet+: An improved deep learning label fusion method for multi-atlas segmentation. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 363–367. IEEE, 2020.

[14] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[15] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

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

[17] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[18] GŘnther Grabner, Andrew L Janke, Marc M Budge, David Smith, Jens Pruessner, and D Louis Collins. Symmetric atlasing and model based segmentation: an application to the hippocampus in older adults. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 58–66. Springer, 2006.

[19] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[20] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[21] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[22] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Rolf A Heckemann, Joseph V Hajnal, Paul Aljabar, Daniel Rueckert, and Alexander Hammers. Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. *NeuroImage*, 33(1):115–126, 2006.

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

[26] Juan Eugenio Iglesias and Mert R Sabuncu. Multi-atlas segmentation of biomedical images: a survey. *Medical image analysis*, 24(1):205–219, 2015.

[27] Ozan Irsoy and Ethem Alpaydin. Autoencoder trees. In *Asian Conference on Machine Learning*, pages 378–390, 2016.

[28] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers

tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19, 2017.

[29] Rohit Jena and Suyash P Awate. A Bayesian neural net to segment images with uncertainty estimates and good calibration. In *International Conference on Information Processing in Medical Imaging*, pages 3–15. Springer, 2019.

[30] Byeongmoon Ji, Hyemin Jung, Jihyeun Yoon, Kyungyul Kim, and Younghak Shin. Bin-wise temperature scaling (BTS): Improvement in confidence calibration performance through simple scaling techniques. *arXiv preprint arXiv:1908.11528*, 2019.

[31] Davood Karimi and Ali Gholipour. Improving calibration and out-of-distribution detection in medical image segmentation with convolutional neural networks. *arXiv preprint arXiv:2004.06569*, 2020.

[32] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5580–5590, 2017.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12295–12305, 2019.

[35] Meelis Kull, Telmo M Silva Filho, Peter Flach, et al. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052–5080, 2017.

[36] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814, 2018.

[37] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[38] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[39] Tim Leathart, Eibe Frank, Geoffrey Holmes, and Bernhard Pfahringer. Probability calibration trees. In *Asian Conference on Machine Learning*, pages 145–160. PMLR, 2017.

[40] Chen-Yu Lee, Patrick Gallagher, and Zhuowen Tu. Generalizing pooling functions in CNNs: Mixed, gated, and tree. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):863–875, 2017.

[41] Zeju Li, Konstantinos Kamnitsas, and Ben Glocker. Overfitting of neural nets under class imbalance: Analysis and improvements for segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 402–410. Springer, 2019.

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[43] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[44] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13132–13143, 2019.

[45] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[46] Juan Maroñas, Roberto Paredes, and Daniel Ramos. Calibration of deep probabilistic models with decoupled Bayesian neural networks. *Neurocomputing*, 2020.

[47] Alireza Mehrtash, William M Wells III, Clare M Tempany, Purang Abolmaesumi, and Tina Kapur. Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *arXiv preprint arXiv:1911.13273*, 2019.

[48] Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based Gaussian processes for large-scale calibrated classification. *arXiv preprint arXiv:1805.10915*, 2018.

[49] Marc Modat, David M Cash, Pankaj Daga, Gavin P Winston, John S Duncan, and Sébastien Ourselin. Global image registration using a symmetric block-matching approach. *Journal of Medical Imaging*, 1(2):024003, 2014.

[50] Marc Modat, Gerard R Ridgway, Zeike A Taylor, Manja Lehmann, Josephine Barnes, David J Hawkes, Nick C Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Computer methods and programs in biomedicine*, 98(3):278–284, 2010.

[51] Azadeh Sadat Mozafari, Hugo Siqueira Gomes, Wilson Leão, Steeven Janny, and Christian Gagné. Attended temperature scaling: A practical approach for calibrating deep neural networks. *arXiv preprint arXiv:1810.11586*, 2018.

[52] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. Calibrating deep neural networks using focal loss. *arXiv preprint arXiv:2002.09437*, 2020.

[53] Allan H Murphy and Robert L Winkler. Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 26(1):41–47, 1977.

[54] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[55] Mahdi Pakdaman Naeini and Gregory F Cooper. Binary classifier calibration using an ensemble of near isotonic regression models. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 360–369. IEEE, 2016.

[56] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005.

[57] Jeremy Nixon, Mike Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. *arXiv preprint arXiv:1904.01685*, 2019.

[58] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[59] Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Byron Boots, and Richard Hartley. Intra order-preserving functions for calibration of multi-class neural networks. *arXiv preprint arXiv:2003.06820*, 2020.

[60] Daniel Rueckert, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach, and David J Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE transactions on medical imaging*, 18(8):712–721, 1999.

[61] Mert R Sabuncu, BT Thomas Yeo, Koen Van Leemput, Bruce Fischl, and Polina Golland. A generative model for image segmentation based on label fusion. *IEEE transactions on medical imaging*, 29(10):1714–1729, 2010.

[62] David W Shattuck, Mubeena Mirza, Vitria Adisetiyo, Cornelius Hojatkashani, Georges Salamon, Katherine L Narr, Russell A Poldrack, Robert M Bilder, and Arthur W Toga. Construction of a 3D probabilistic atlas of human cortical structures. *Neuroimage*, 39(3):1064–1080, 2008.

[63] Gia-Lac Tran, Edwin V Bonilla, John Cunningham, Pietro Michiardi, and Maurizio Filippone. Calibrating deep convolutional gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1554–1563. PMLR, 2019.

[64] Hongzhi Wang, Jung W Suh, Sandhitsu R Das, John B Pluta, Caryne Craige, and Paul A Yushkevich. Multi-atlas segmentation with joint label fusion. *IEEE transactions on pattern analysis and machine intelligence*, 35(3):611–623, 2012.

[65] Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. Non-parametric calibration for classification. In *International Conference on Artificial Intelligence and Statistics*, pages 178–190. PMLR, 2020.

[66] Long Xie, Jiancong Wang, Mengjin Dong, David A Wolk, and Paul A Yushkevich. Improving multi-atlas segmentation by convolutional neural network based patch error estimation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 347–355. Springer, 2019.

[67] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *ICML*, volume 1, pages 609–616. Citeseer, 2001.

[68] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[69] Jize Zhang, Bhavya Kailkhura, and T Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. *arXiv preprint arXiv:2003.07329*, 2020.

Local Temperature Scaling for Probability Calibration
Supplementary Material

This supplementary material provides additional details for our approach. Specifically,

1. Appx. A briefly introduces additional related work about uncertainty quantification. This section connects with §2 in the main manuscript and provides additional comments regarding uncertainty quantification approaches in relation to our approach.

2. Appx. B describes the networks we use for LTS and IBTS. This section connect with §3.4 and Fig. 3 in the main manuscript and provides details about the tree-like convolutional neural network we use to train the IBTS and LTS models. We emphasize that the network architecture is not our contribution, it is inspired and modified from [40] and other network architectures could also work.

3. Appx. C provides dataset descriptions and implementation details. This section connects with §4, §4.1, §4.2, §4.3, and §4.4 in the main manuscript and details (1) the dataset we use; (2) the training/validation/testing data split of segmentation and calibration; (3) the specific hyper-parameters we use to train both segmentation models and calibration models; (4) the GitHub repositories for baseline calibration methods we compare against.

4. Appx. D provides additional examples for local reliability diagrams. This section connects with §3.4 and Fig. 2 in the main manuscript to additionally show the spatially-variant feature of our LTS approach.

5. Appx. E discusses our temperature scaling approaches from an entropy point of view. This section connects with §3.5 in the main manuscript to prove the theorems to support our claims. Specifically, this section discusses the relation of entropy and cross entropy and uncovers why our temperature scaling approaches (TS, IBTS, LTS) works.

6. Appx. F details the evaluation metrics we use for semantic segmentation. This section connects with §4, Fig. 2, Tab. 1, and Tab. 2 in the main manuscript to provide formal definitions for all our evaluation measures.

7. Appx. G illustrates the *Boundary* and *All* evaluation regions. This section connects with §4 and Tab. 1 in the main manuscript to illustrate a visual example of the different regions we evaluate. Note that the results in the *All* region reflect the overall calibration performance for an image segmentation; results in the *Boundary* region reflect the most challenging calibration performance for an image segmentation.

8. Appx. H shows evaluation results for the *Local* region for different patch sizes. This section connects with §4 and Tab. 1 in the main manuscript to indicate how the local patch size influences the quantitative results. Note that results in the *Local* region generally reflect whether the calibration method can handle spatial variations. This is different from the *All* and *Boundary* regions discussed in Appx. G above.

9. Appx. I discusses variations across the different datasets. This section connects with §4.1, §4.2, §4.3, §4.4 and Tab. 1 in the main manuscript and explains the different magnitudes of the quantitative results for different datasets. Specifically, the COCO dataset shows the biggest variantions, followed by the CamVid dataset and lastly LPBA40 exhibits the smallest variations. Due to the different levels of variation of the different datasets, the reported values in COCO are larger than those in CamVid and the smallest values are observed in LPBA40.

10. Appx. J contains additional evaluation results besides the results presented in Tab. 1. This section connects with §4.2 and Tab. 1 in the main manuscript to further strengthen our manuscript. These results are line with the conclusions we obtain in §4, i.e. our LTS approach generally works best among different baseline methods.

11. Appx. K provides details on joint label fusion for multi-atlas segmentation. This section connects with §4.4 and Tab. 2 in the main manuscript to provide details about the downstream MAS label fusion task. Specifically, this section illustrates why the VoteNet+ based joint label fusion method is sensitive to accurate probability predictions, which in turn demonstrates that improved calibration of our approach results in improved fused segmentation results.

A. Additional Related Work

Probability calibration can be used for uncertainty estimation [37] as calibrated probabities can directly be used as measures of uncertainty. However, methods that provide uncertainty estimates are not necessarily calibrated. Most existing work on uncertainty estimation starts with a Bayesian formulation [37, 29, 46], whereby a prior distribution is specified, and the posterior distribution over the parameters is optimized over the training data. These Bayesian models should result in better calibrated probability measures if their prior assumptions are valid. However, when some of the underlying assumptions are violated, the results may not be calibrated: [32] is a good example for a Bayesian model improving calibration, but not achieving it. Other uncertainty estimation approaches include ensembles [37] and Monte Carlo dropout [16], which help probability calibration but do not directly cope or achieve it. Gaussian Process (GP) approaches [65] can inherently provide good uncertainty estimates, but may suffer from lower accuracy and higher computational complexity on high-dimensional classification tasks. In particular, a GP will only provide calibrated measures of uncertainty if the Gaussian assumption is valid. In practice, this may not be the case when combined with a deep network [63]. Further, GP models are costly for classification and GP regression formulations require calibration [48, 65]. Our formulation is entirely different and directly predicts calibration parameters for softmax layers. Our model does not depend on any assumption and is a completely poct-hoc approach for any pre-trained segmentation model with probability outputs.

B. Networks for LTS and IBTS

To obtain T^* in Eq. (3.3), we directly optimize the parameter T with respect to the NLL loss on the hold-out validation dataset.

To obtain $T_i(x)^*$, we borrow the idea of soft decision trees [27] and propose to use a tree-like convolutional neural network [40] to predict $T_i(x)$, which has fewer parameters than a standard convolutional neural network while achieving comparable state-of-the-art performance [40]. We resort to such a simpler tree-like model, because one of the datasets that we use for evaluation is relatively small, though more complex models could be further explored.

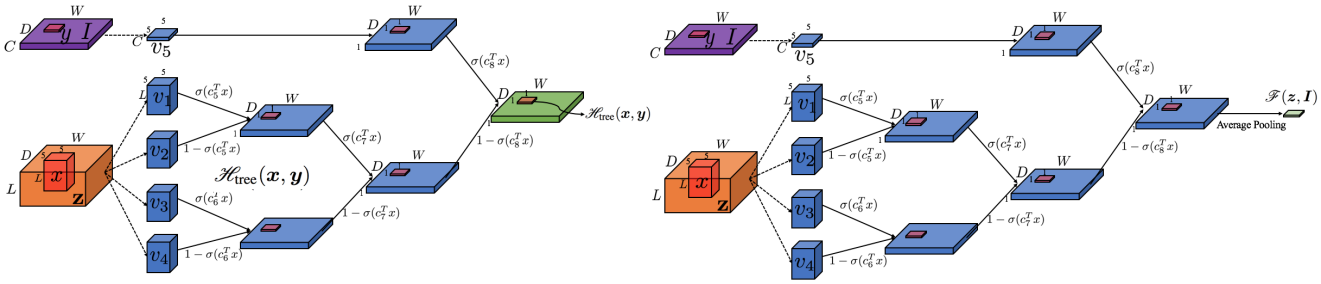

Figure 4: LTS (left) and IBTS (right) hierarchical tree-like architectures demonstrated in 2-D. W is the image width, D is the image length, L is the number of classes, C is the number of channels. $\boldsymbol{x}$ is the patch centered at location x of size $L \times 5 \times 5$. Its corresponding patch inside image I is denoted by $\boldsymbol{y}$, which is of size $C \times 5 \times 5$. σ is the sigmoid function. Input to the model are the logits of size $L \times W \times D$. Output is the spatially varying temperature value of the image ($1 \times W \times D$) for LTS or an image-dependent temperature scalar value ($1 \times 1 \times 1$) for IBTS. $\boldsymbol{v}_i$ and $\boldsymbol{c}_j$ are convolutional filters of size $L \times 5 \times 5$ (except $\boldsymbol{v}_5$ is of size $C \times 5 \times 5$ to be compatible with the size of image). Note that the dilation is 2 for all convolutional filters, thus resulting in a 9×9 receptive field.

The proposed framework is constructed as a pre-specified hierarchical binary tree in which each leaf is a convolutional filter learned during training. Denote the leaf node with index m as $\boldsymbol{v}_m$, the patch in logits $\mathbf{z}$ to be convolved as $\boldsymbol{x}$ and its corresponding patch in image $\mathbf{I}$ to be convolved as $\boldsymbol{y}$. Since a convolutional layer can be transformed into a fully-connected layer, which is essentially a matrix multiplication plus a bias offset, we use $\boldsymbol{v}_m^T\boldsymbol{x}$ to represent the convolution operation in the framework for ease of notation (omit bias offset for simplicity). For internal nodes of the tree, each parent node value is a mixture (i.e. weighted average) of children nodes' values and the mixture parameter is also learned during training. Specifically, we use a convolution operation $\boldsymbol{c}_m$ plus a sigmoid function σ to determine the mixture parameter $\sigma(\boldsymbol{c}_m^T\boldsymbol{x})$. The root node is the final output. For IBTS, the output is a single temperature value for the logits, while, for LTS, the output is a temperature map which has the same size as the input logits, except that the number of feature channels is 1. Thus, the nodes

of the tree can be represented as follows:

$$\mathscr{H}_m(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \boldsymbol{v}_m^T\boldsymbol{y}+1 & \text{if leaf node in image} \\ \boldsymbol{v}_m^T\boldsymbol{x}+1 & \text{if leaf node in logits} \\ \sigma(\boldsymbol{c}_m^T\boldsymbol{x})\mathscr{H}_{m,\text{logits,left}}(\boldsymbol{x}) + (1-\sigma(\boldsymbol{c}_m^T\boldsymbol{x}))\mathscr{H}_{m,\text{logits,right}}(\boldsymbol{x}) & \text{if internal node in logits} \\ \text{ReLU}\big(\sigma(\boldsymbol{c}_m^T\boldsymbol{x})\mathscr{H}_{m,\text{logits}}(\boldsymbol{x}) + (1-\sigma(\boldsymbol{c}_m^T\boldsymbol{x}))\mathscr{H}_{m,\text{image}}(\boldsymbol{y})\big)+\varepsilon & \text{if root node} \end{cases}, \tag{B.1}$$

where ReLU is the Rectified Linear Unit, $\mathscr{H}_m(\boldsymbol{x},\boldsymbol{y})$ is the root node value, $\mathscr{H}_{m,\text{logits,left}}(\boldsymbol{x})$ and $\mathscr{H}_{m,\text{logits,right}}(\boldsymbol{x})$ are the left child node value and right child node value for internal nodes in logits, respectively. $\mathscr{H}_{m,\text{logits}}(\boldsymbol{x})$ is the top node containing information only from the logits and $\mathscr{H}_{m,\text{image}}(\boldsymbol{y})$ is the top node containing information only from the image. ε is a very small positive real number to guarantee the positivity for the output temperature value. The $+1$ value for the leaf node is for model initialization and stabilization. With this trick, the learning process is more stable and the performance is much better. If there are only leaf nodes, then the convolution filters are trying to learn the residual of the temperature scalar value with respect to the standard uncalibrated temperature value 1. Fig. 4(left) illustrates the proposed tree-like learning framework for LTS. For simplicity, let us assume the output is positive, then the specific representation becomes

$$\begin{aligned} \mathscr{H}_{\text{tree}}(\boldsymbol{x},\boldsymbol{y}) = {} & \sigma(\boldsymbol{c}_8^T\boldsymbol{x})(\boldsymbol{v}_5^T\boldsymbol{y}+1) \\ & + (1-\sigma(\boldsymbol{c}_8^T\boldsymbol{x}))\big\{\sigma(\boldsymbol{c}_7^T\boldsymbol{x})\big[\sigma(\boldsymbol{c}_5^T\boldsymbol{x})(\boldsymbol{v}_1^T\boldsymbol{x}+1)+(1-\sigma(\boldsymbol{c}_5^T\boldsymbol{x}))(\boldsymbol{v}_2^T\boldsymbol{x}+1)\big] \\ & + (1-\sigma(\boldsymbol{c}_7^T\boldsymbol{x}))\big[\sigma(\boldsymbol{c}_6^T\boldsymbol{x})(\boldsymbol{v}_3^T\boldsymbol{x}+1)+(1-\sigma(\boldsymbol{c}_6^T\boldsymbol{x}))(\boldsymbol{v}_4^T\boldsymbol{x}+1)\big]\big\}. \end{aligned} \tag{B.2}$$

To connect back to the definition in §3.4, $\mathscr{H}_{\text{tree}}$ is the network $\mathscr{H}$, $\boldsymbol{v}_i$ and $\boldsymbol{c}_j$ are parameters α, $\boldsymbol{x}$ is the patch centered at location x in logits $\mathbf{z}$, $\boldsymbol{y}$ is the corresponding patch of image $\mathbf{I}$.

To obtain T_i^*, we modify the above-mentioned network $\mathscr{H}_{\text{tree}}$ to predict one temperature value T_i for each image. We add an average pooling layer after $\mathscr{H}_{\text{tree}}$ to get the image-based temperature value. Specifically, using $\mathscr{F}$ to represent the network of IBTS as in Eq. (3.5), we have $\mathscr{F} = \frac{1}{|\Omega|}\sum_{x\in\Omega}\mathscr{H}_{\text{tree}}(\boldsymbol{x},\boldsymbol{y})$, where $\boldsymbol{x}$ is the patch centered at location x in logits $\mathbf{z}$, $\boldsymbol{y}$ is the corresponding batch of $\boldsymbol{x}$ in image $\mathbf{I}$, and Ω is the logits space. Fig. 4(right) illustrates the proposed tree-like learning framework for IBTS. Source code is publicly-available at https://github.com/uncbiag/LTS.

C. Dataset Description and Implementation Details

We use the following image segmentation datasets in our experiments:

1. **COCO** [42]: The Common Object in Context (COCO) [42] dataset is a large-scale dataset of complex images. It provides pixel-level labels for 118K training images (COCO train2017) and 5K validation images (COCO val2017). Further, the COCO-stuff [8] dataset augments COCO with dense pixel-level annotations for 80 thing classes and 91 stuff classes. For simplicity, we focus on the 20 categories that are present in the Pascal VOC [14] dataset for our experiments, considering the remaining classes as background.

2. **CamVid** [7, 6]: The Cambridge-driving Labeled Video Database (CamVid) [7, 6] is a collection of videos with object class semantic labels. We use the split and image resolution as in [28], which consists of 367 frames for training, 101 frames for validation and 233 frames for testing. Each frame has a size of 360×480 and its pixels are labeled with 11 semantic classes excluding background.

3. **LPBA40** [62]: The LONI Probabilistic Brain Atlas (LPBA40) [62] dataset contains 40 T1-weighted 3D brain MR images from healthy patients. Each image has labels for 56 manually segmented structures. For preprocessing, all images are first affinely registered to the ICBM MNI152 nonlinear atlas [18] using NiftyReg [49, 50, 60] and intensity normalized via histogram equalization.

For the Fully-Convolutional Network (FCN) experiment in §4.1, we use the COCO val2017 dataset for our calibration experiment in which the training/validation/testing images are partitioned in sets of size 3.5K/0.5K/1K, respectively. We use the PyTorch pre-trained model[1] for semantic segmentation on the COCO dataset. This is an FCN [43] with a ResNet-101 [23]

[1] https://pytorch.org/docs/stable/torchvision/models.html#semantic-segmentation

backbone. The pre-trained model has been trained on a subset of COCO train2017, i.e., for the 20 categories that are present in the Pascal VOC [14] dataset. For details, please resort to the Pytorch official webpage (footnote) mentioned above.

For the Tiramisu experiment in §4.2, we use the hold-out validation dataset for our calibration experiment in which the training/validation images are 90/11. Finally the calibration performance is tested on the testing dataset which includes 233 images. We use the PyTorch Tiramisu[2] segmentation model [28] on the CamVid dataset. Training details are included in the GitHub repository.

For the U-Net experiment in §4.3, we use a 2-fold cross-validation setup to cover all the 40 images in the dataset. Training/validation/testing images are partitioned as 17/3/20. This is consistent with the setting in [12]. We use 4-fold cross-validation for our calibration experiment to cover all 40 images. Training/validation/testing images are partitioned as 10/3/10 for each fold. The U-Net takes patches of $72 \times 72 \times 72$ of the training images, where the $40 \times 40 \times 40$ patch center is used to tile the volume. The output is the voxel-wise probability of each label at each position. Training patches are randomly cropped assuring at least 5% correct labels in the patch volume. We use `Adam` [33] with 300 epochs and a multi-step learning rate. The initial learning rate is 1e-3, and then reduced by 90% at the 150-th epoch and the 250-th epoch, respectively. Cross-entropy loss is used as the loss function. When calibrating, within each fold of the U-Net 2-fold cross validation, we perform another 2-fold cross validation. Specifically, 23 images (3 from validation and 20 from testing) are split into 10/3/10 for train/validation/test. 2-fold cross-validation will cover all 20 testing images of U-Net testing. This design results in a 4-fold cross validation experiment to cover all 40 images.

For the Downstream MAS label fusion experiment in §4.4, we use 2-fold cross-validation to cover all the images. In each fold, 17 atlases are chosen. Training/validation/testing images are partitioned as 272/51/340. This is consistent with the setting in [13]. We use 4-fold cross-validation for the calibration experiments to cover all images. Training/validation/testing are partitioned as 170/51/170 for each fold. Training data for VoteNet+ is acquired by deformable image registrations. Specifically, the same 17 images as for the U-Net training are chosen as atlas images. First, all 17 atlases are registered to each other, which results in $17 \times 16 = 272$ pairs of training data. Then all 17 atlases are registered to the 3 validation images for the U-Net, which results in $17 \times 3 = 51$ pairs of validation data. Finally, all 17 atlases are registered to the 20 testing images for the U-Net, which results in $17 \times 20 = 340$ pairs of testing data. The same 2-fold cross-validation strategy still applies to VoteNet+, but with the data split as 272/51/340 for train/validation/test. VoteNet+ takes patches of $72 \times 72 \times 72$ from the target image and a warped atlas image at the same position, where the $40 \times 40 \times 40$ patch center is used to tile the volume. The output is the voxel-wise probability, indicating whether the warped atlas label is equal to the target image label. We use `Adam` [33] with 500 epochs with a multi-step learning rate. The initial learning rate is 1e-3 and then reduced by half at the 200-th epoch, 350-th epoch, and 450-th epoch respectively. Same as for the U-Net, training patches are randomly cropped assuring at least 5% correct labels in the patch volume. Binary cross-entropy is used as the loss function. When calibrating, within each fold of the VoteNet+ 2-fold cross validation, we perform a 2-fold cross validation. Specifically, 391 pairs (51 from validation and 340 from testing) are split into 170/51/170 for train/validation/test. 2-fold cross-validation will cover all 340 testing pairs of VoteNet+ testing. This design results in a 4-fold cross validation experiment to cover all 680 pairs. Furthermore, we use joint label fusion (JLF) [64] to obtain the final segmentation for each image. See Appx. K for more information on MAS and label fusion, as well as experimental details.

To train IBTS and LTS, we use `Adam` [33] with 100 epochs and a multi-step learning rate. The initial learning rate for the LPBA40 dataset is 1e-4 and is reduced to 1e-5 after 50 epochs, while for the COCO and the CamVid dataset, it is 1e-5 and is reduced to 1e-6 after 50 epochs. We use the cross-entropy loss. The loss is evaluated over the *All* region to ignore the majority of the background.

The FL and MMCE losses are from the GitHub repository[3] of [52]. Isotonic regression (IsoReg) [68] and ensemble temperature scaling (ETS) [69] are from the GitHub repository[4] of [69]. Vector scaling (VS) [20] and Dirichlet calibration with off-diagonal regularization (DirODIR) [34] are from the GitHub repository[5] of [34]. Training with FL and MMCE follows the same recipe as training with the multi-class entropy loss except that the training loss term is changed. The GitHub imple-

[2] The implementation follows this GitHub repository: `https://github.com/bfortuner/pytorch_tiramisu`
[3] `https://github.com/torrvision/focal_calibration/tree/main/Losses`
[4] `https://github.com/zhang64-llnl/Mix-n-Match-Calibration`
[5] `https://github.com/dirichletcal/experiments_neurips`

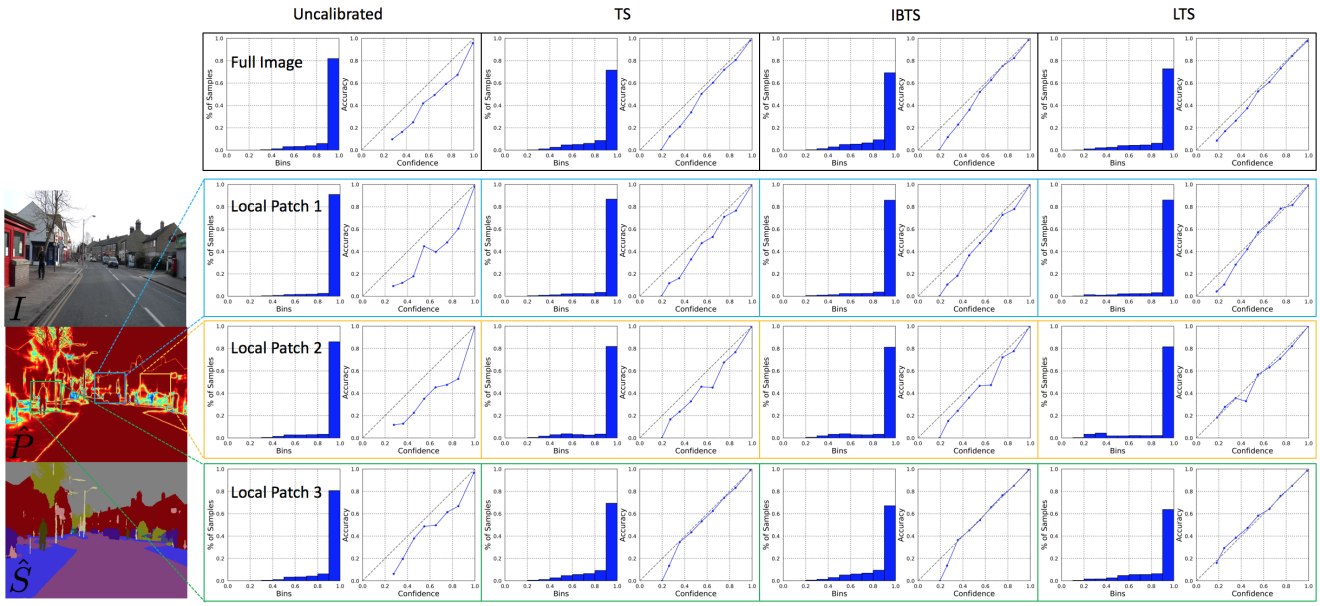

Figure 5: An example of global and local reliability diagrams for different methods for the Tiramisu semantic segmentation experiment (§4.2). I is the image, $\hat{P}$ is the predicted uncalibrated probability, and $\hat{S}$ is the predicted segmentation. Figures are displayed in couples, where the left figure is the probability distribution of pixels/voxels while the right figure is the reliability diagram (See Appx. F for definitions). The top row shows the global reliability diagrams for different methods for the entire image. The three rows underneath correspond to local reliability diagrams for the different methods for different local patches. LTS not only calibrates probabilities well for the entire image but also calibrates probabilities better than TS and IBTS in local pacthes.

mentation repository[6] provides all details about the hyper-parameters of training of the deep Tiramisu network; we thus omit them here to avoid duplication. For DirODIR, the hyper-parameters for off-diagonal regularization and bias regularization are both set to 0.01. We use `Adam` for a maximum of 100 epochs with early stop patience set to 10 epochs, i.e. training stops early if 10 consecutively worse epochs are observed. The model is trained with an initial learning rate of 1e-3 and fine-tuned with a learning rate of 1e-4.

D. Local Reliability Diagrams

To visualize the spatially-varying property of LTS, we show the local reliability diagram of Tiramisu for the CamVid experiment in Fig. 5. Similar to the conclusion from Fig. 2, Fig. 5 also suggests that LTS performs better than TS and IBTS for the entire image as well as for the local image patches. This observation is consistent with results in Tab. 1.

E. Temperature Scaling from Entropy Point of View

Temperature scaling can also be connected to entropy [20]. In this section, we establish the relation between entropy and temperature scaling by showing that different temperature scaling models are indeed the solutions for entropy maximization or minimization subject to different constraints. Note that a related insight has been proposed in [20] for classification. We extend it to semantic segmentation for our different temperature scaling settings and provide detailed discussions. Specifically, we show the solutions of TS, IBTS and LTS when minimizing NLL in Appx. E.1; we define overconfidence and underconfidence in Appx. E.2; we show the entropy maximization and minimization solutions without constraints in Appx. E.3; we deduct the solutions for entropy maximization under the condition of overconfidence as well as for entropy minimization under the condition of underconfidence in Appx. E.4; finally, we show that the solutions for minimizing NLL w.r.t. TS, IBTS, LTS are also the solutions for entropy maximization in the case of overconfidence or the solutions for entropy minimization in the case of underconfidence in Appx. E.5. Overall, TS, IBTS and LTS determined based on a given dataset results in NLL (cross entropy) and entropy reaching an equilibrium which empirically corresponds to a well-calibration state.

[6] https://github.com/bfortuner/pytorch_tiramisu

E.1. Minimize NLL with (Local) Temperature Scaling

Lemma 1. *Given a logit vector map $\boldsymbol{z}(x)$ at position x and its corresponding probability map obtained via softmax function (σ_{SM}) the weighted averaged logits with temperature scaling (TS) are (1) monotonic with respect to temperature value and (2) yield the following bounds:*

$$\frac{1}{L}\sum_{l=1}^{L}\boldsymbol{z}(x)^{(l)} \leq \sum_{l=1}^{L}\boldsymbol{z}(x)^{(l)}\sigma_{SM}\big(\boldsymbol{z}(x)/T\big)^{(l)} \leq \max_{l}\{\boldsymbol{z}(x)^{(l)}\}. \tag{E.1}$$

Proof. Let $\lambda = \frac{1}{T}$ and denote $\mathcal{F}(\lambda) = \sum_{l=1}^{L}\mathbf{z}(x)^{(l)}\sigma_{SM}\big(\lambda\mathbf{z}(x)\big)^{(l)} = \sum_{l=1}^{L}\mathbf{z}(x)^{(l)}\frac{\exp\left(\lambda\mathbf{z}(x)^{(l)}\right)}{\sum_{j=1}^{L}\exp\left(\lambda\mathbf{z}(x)^{(j)}\right)}$. Then we take the derivative with respect to λ,

$$\frac{\partial\mathcal{F}(\lambda)}{\partial\lambda} = \frac{\Big(\sum_{l=1}^{L}(\mathbf{z}(x)^{(l)})^2\exp\left(\lambda\mathbf{z}(x)^{(l)}\right)\Big)\Big(\sum_{l=1}^{L}\exp\left(\lambda\mathbf{z}(x)^{(l)}\right)\Big) - \Big(\sum_{l=1}^{L}\mathbf{z}(x)^{(l)}\exp\left(\lambda\mathbf{z}(x)^{(l)}\right)\Big)^2}{\Big(\sum_{j=1}^{L}\exp\left(\lambda\mathbf{z}(x)^{(j)}\right)\Big)^2}. \tag{E.2}$$

By the Cauchy–Schwarz inequality, we have

$$\Big(\sum_{l=1}^{L}(\mathbf{z}(x)^{(l)})^2\exp\left(\lambda\mathbf{z}(x)^{(l)}\right)\Big)\Big(\sum_{l=1}^{L}\exp\left(\lambda\mathbf{z}(x)^{(l)}\right)\Big) \geq \Big(\sum_{l=1}^{L}\mathbf{z}(x)^{(l)}\exp\left(\lambda\mathbf{z}(x)^{(l)}\right)\Big)^2.$$

Thus, $\frac{\partial\mathcal{F}(\lambda)}{\partial\lambda} \geq 0$. This indicates that the function $\mathcal{F}(\lambda)$ is monotonicly increasing with respect to λ. Since the temperature scaling value T is non-negative, i.e., $T \in \mathbb{R}^+$, we have $\lambda \in \mathbb{R}^+$. Furthermore,

$$\begin{aligned} \lambda \to 0, \quad & \sigma_{SM}\big(\lambda\mathbf{z}(x)\big)^{(l)} = \frac{1}{L}, \quad \forall l = 1, ..., L; \\ \lambda \to +\infty, \quad & \sigma_{SM}\big(\lambda\mathbf{z}(x)\big)^{(l)} = \begin{cases} 1, & \max_j\{\mathbf{z}(x)^{(j)}\} = \mathbf{z}(x)^{(l)}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{E.3}$$

Therefore, we have $\frac{1}{L}\sum_{l=1}^{L}\mathbf{z}(x)^{(l)} \leq \mathcal{F}(\lambda) \leq \max_l\{\mathbf{z}(x)^{(l)}\}$.

□

Remark. If T is allowed to be negative, i.e. $T \in \mathbb{R}$, then the following bounds hold:

$$\min_{l}\{\mathbf{z}(x)^{(l)}\} \leq \sum_{l=1}^{L}\mathbf{z}(x)^{(l)}\sigma_{SM}\big(\mathbf{z}(x)/T\big)^{(l)} \leq \max_{l}\{\mathbf{z}(x)^{(l)}\}. \tag{E.4}$$

Theorem 1. *Given n logit vector maps $\boldsymbol{z}_1, ..., \boldsymbol{z}_n$ and label maps $S_1, ..., S_n$, the optimal temperature values of temperature scaling (TS), image-based temperature scaling (IBTS) and local temperature scaling (LTS) to the following NLL minimization problem*

$$\begin{aligned} \min_{\alpha_i(x)} & -\sum_{i=1}^{n}\sum_{x\in\Omega}\log\Big(\sigma_{SM}\big(\alpha_i(x)\boldsymbol{z}_i(x)\big)^{(S_i(x))}\Big) \\ & \textit{subject to} \quad \alpha_i(x) \geq 0 \end{aligned} \tag{E.5}$$

are

$$\begin{cases} \alpha^* = 0, & \textit{if} \quad \sum_{i=1}^{n}\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)} \\ \Big\{\alpha^* > 0 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha^*\boldsymbol{z}_i(x)\big)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise} \end{cases}$$

$$\begin{cases} \alpha_i^* = 0, & \textit{if} \quad \sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)} \\ \Big\{\alpha_i^* > 0 \mid \sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha_i^*\boldsymbol{z}_i(x)\big)^{(l)} = \sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise} \end{cases} \tag{E.6}$$

$$\begin{cases} \alpha_i(x)^* = 0, & \textit{if} \quad \boldsymbol{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)} \\ \Big\{\alpha_i(x)^* > 0 \mid \sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha_i(x)^*\boldsymbol{z}_i(x)\big)^{(l)} = \boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise} \end{cases},$$

where

$$\begin{aligned}
&\textit{(TS):} \quad \alpha_i(x) \coloneqq \alpha, \forall i, x, \quad \textit{and} \quad T \coloneqq \frac{1}{\alpha}, T \in \mathbb{R}^+ \\
&\textit{(IBTS):} \quad \alpha_i(x) \coloneqq \alpha_i, \forall x, \quad \textit{and} \quad T_i \coloneqq \frac{1}{\alpha_i}, T_i \in \mathbb{R}^+ \\
&\textit{(LTS):} \quad \alpha_i(x) \coloneqq \alpha_i(x), \quad \textit{and} \quad T_i(x) \coloneqq \frac{1}{\alpha_i(x)}, T_i(x) \in \mathbb{R}^+.
\end{aligned} \tag{E.7}$$

Proof. For TS, Let

$$\mathcal{F}(\alpha) = -\sum_{i=1}^{n}\sum_{x\in\Omega} \log\Big(\sigma_{SM}\big(\alpha \mathbf{z}_i(x)\big)^{(S_i(x))}\Big). \tag{E.8}$$

Taking the derivative w.r.t. α we obtain

$$\frac{\partial \mathcal{F}(\alpha)}{\partial \alpha} = -\sum_{i=1}^{n}\sum_{x\in\Omega} \Big(\mathbf{z}_i(x)^{(S_i(x))} - \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \sigma_{SM}\big(\alpha \mathbf{z}_i(x)\big)^{(l)}\Big) \tag{E.9}$$

Case 1: If $\sum_{i=1}^{n}\sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$, we have $\frac{\partial \mathcal{F}(\alpha)}{\partial \alpha} \mid_{\alpha=0} \geq 0$. With Lemma 1, $\mathcal{F}(\alpha)$ is a monotonic increasing function. This indicates the minimum value is achieved at $\alpha = 0$.
Case 2: If $\sum_{i=1}^{n}\sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$. With Lemma 1 we have $\frac{\partial \mathcal{F}(\alpha)}{\partial \alpha} \mid_{\alpha=0} < 0$ and $\frac{\partial \mathcal{F}(\alpha)}{\partial \alpha} \mid_{\alpha\to+\infty} \geq 0$. From the intermediate value theorem and Lemma 1, we know there exists a unique α^* ($\{\alpha^* > 0 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \sigma_{SM}\big(\alpha^* \mathbf{z}_i(x)\big)^{(j)} = \sum_{i=1}^{n}\sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))}\}$) such that $\frac{\partial \mathcal{F}(\alpha)}{\partial \alpha} \mid_{\alpha=\alpha^*} = 0$. This α^* is the point where $\mathcal{F}(\alpha)$ reaches the minimum value.

For IBTS, let

$$\mathcal{F}(\alpha_i) = -\sum_{i=1}^{n}\sum_{x\in\Omega} \log\Big(\sigma_{SM}\big(\alpha_i \mathbf{z}_i(x)\big)^{(S_i(x))}\Big). \tag{E.10}$$

Taking the derivative w.r.t. α_i, we obtain

$$\frac{\partial \mathcal{F}(\alpha_i)}{\partial \alpha_i} = -\sum_{x\in\Omega} \Big(\mathbf{z}_i(x)^{(S_i(x))} - \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \sigma_{SM}\big(\alpha_i \mathbf{z}_i(x)\big)^{(l)}\Big), \quad \forall i. \tag{E.11}$$

Case 1: If $\sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$, we have $\frac{\partial \mathcal{F}(\alpha_i)}{\partial \alpha_i} \mid_{\alpha_i=0} \geq 0$. With Lemma 1, $\mathcal{F}(\alpha_i)$ is a monotonic increasing function. This indicates the minimum value is achieved at $\alpha_i = 0$.
Case 2: If $\sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$. With Lemma 1 we have $\frac{\partial \mathcal{F}(\alpha_i)}{\partial \alpha_i} \mid_{\alpha_i=0} < 0$ and $\frac{\partial \mathcal{F}(\alpha_i)}{\partial \alpha_i} \mid_{\alpha_i\to+\infty} \geq 0$. From the intermediate value theorem and Lemma 1, we know there exists a unique α_i^* ($\{\alpha_i^* > 0 \mid \sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \sigma_{SM}\big(\alpha_i^* \mathbf{z}_i(x)\big)^{(j)} = \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))}\}$) such that $\frac{\partial \mathcal{F}(\alpha_i)}{\partial \alpha_i} \mid_{\alpha_i=\alpha_i^*} = 0$. This α_i^* is the point where $\mathcal{F}(\alpha_i)$ reaches the minimum value.

For LTS, let

$$\mathcal{F}(\alpha_i(x)) = -\sum_{i=1}^{n}\sum_{x\in\Omega} \log\Big(\sigma_{SM}\big(\alpha_i(x) \mathbf{z}_i(x)\big)^{(S_i(x))}\Big). \tag{E.12}$$

Taking the derivative w.r.t. $\alpha_i(x)$, we obtain

$$\frac{\partial \mathcal{F}(\alpha_i(x))}{\partial \alpha_i(x)} = -\Big(\mathbf{z}_i(x)^{(S_i(x))} - \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \sigma_{SM}\big(\alpha_i(x) \mathbf{z}_i(x)\big)^{(l)}\Big), \quad \forall i, x. \tag{E.13}$$

Case 1: If $\mathbf{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}$, we have $\frac{\partial \mathcal{F}(\alpha_i(x))}{\partial \alpha_i(x)} \mid_{\alpha_i(x)=0} \geq 0$. With Lemma 1, $\mathcal{F}(\alpha_i(x))$ is a monotonic increasing function. This indicates the minimum value is achieved at $\alpha_i(x) = 0$.
Case 2: If $\mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}$. With Lemma 1 we have $\frac{\partial \mathcal{F}(\alpha_i(x))}{\partial \alpha_i(x)} \mid_{\alpha_i(x)=0} < 0$ and $\frac{\partial \mathcal{F}(\alpha_i(x))}{\partial \alpha_i(x)} \mid_{\alpha_i(x)\to+\infty} \geq 0$. From the intermediate value theorem and Lemma 1, we know there exists a unique $\alpha_i(x)^*$ ($\{\alpha_i(x)^* > 0 \mid \sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha_i(x)^*\mathbf{z}_i(x)\big)^{(j)} = \sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\}$) such that $\frac{\partial \mathcal{F}(\alpha_i(x))}{\partial \alpha_i(x)} \mid_{\alpha_i(x)=\alpha_i(x)^*} = 0$. This $\alpha_i(x)^*$ is the point where $\mathcal{F}(\alpha_i(x))$ reaches the minimum value.

□

Remark. The original temperature scaling method defines T instead of α in Theorem 1. T and α are exchangeable via $T = \frac{1}{\alpha}$. Here we use α to make the proof readable and easy to follow. Furthermore, the definition of temperature scaling requires the temperature value $T > 0$. By using α, we require $\alpha \geq 0$ with $\alpha \to 0$ when $T \to +\infty$.

E.2. Overconfidence and Underconfidence

One indication of overconfidence for semantic segmentation is that the NLL is greater than or equal to the entropy on the testing dataset (and also the validation dataset) (see §3.5 for a detailed explanation). As demonstrated by [52], this greater-than relationship is mainly because the network gradually becomes more and more confident on its incorrect predictions. Mathematically, before calibration, we have the following relationship on the validation (or testing) dataset:

$$-\sum_{i=1}^{n}\sum_{x\in\Omega}\log\Big(\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(S_i(x))}\Big) \geq -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}\log\Big(\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}\Big). \tag{E.14}$$

Furthermore, Eq. (E.14) leads to

$$-\sum_{i=1}^{n}\sum_{x\in\Omega}\Big[\mathbf{z}_i(x)^{(S_i(x))} + \log\Big(\sum_{l=1}^{L}\exp(\mathbf{z}_i(x)^{(l)})\Big)\Big] \geq -\sum_{i=1}^{n}\sum_{x\in\Omega}\Big[\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)} + \underbrace{\sum_{l=1}^{L}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}}_{=1}\log\Big(\sum_{l=1}^{L}\exp(\mathbf{z}_i(x)^{(l)})\Big)\Big] \tag{E.15}$$

$$-\sum_{i=1}^{n}\sum_{x\in\Omega}\Big[\mathbf{z}_i(x)^{(S_i(x))} + \cancel{\log\Big(\sum_{l=1}^{L}\exp(\mathbf{z}_i(x)^{(l)})\Big)}\Big] \geq -\sum_{i=1}^{n}\sum_{x\in\Omega}\Big[\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)} + \cancel{\log\Big(\sum_{l=1}^{L}\exp(\mathbf{z}_i(x)^{(l)})\Big)}\Big] \tag{E.16}$$

$$\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \leq \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}. \tag{E.17}$$

Eq. (E.17) is where the idea of the TS constraint in Eq. (E.40) is coming from. Similarly, if we assume

$$-\sum_{x\in\Omega}\log\Big(\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(S_i(x))}\Big) \geq -\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}\log\Big(\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}\Big) \quad \forall i \tag{E.18}$$

$$-\log\Big(\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(S_i(x))}\Big) \geq -\sum_{l=1}^{L}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}\log\Big(\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}\Big) \quad \forall i, x, \tag{E.19}$$

we get

$$\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \leq \sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}, \quad \forall i \tag{E.20}$$

$$\mathbf{z}_i(x)^{(S_i(x))} \leq \sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\mathbf{z}_i(x)\big)^{(l)}, \quad \forall i, x. \tag{E.21}$$

Hence, Eq. (E.20) is where the idea of the IBTS constraint in Eq. (E.40) is coming from and Eq. (E.21) is where the idea of the LTS constraint in Eq. (E.40) is coming from.

Definition 4. *For semantic segmentation, a model is* ***overconfident*** *for the predicted probabilities in n validation images if*

$$-\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)\leq-\sum_{i=1}^{n}\sum_{x\in\Omega}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$
$$\textit{or} \tag{E.22}$$
$$\sum_{i=1}^{n}\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\leq\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)};$$

a model is ***overconfident*** *for the predicted probabilities in a validation image I_i if*

$$-\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)\leq-\sum_{x\in\Omega}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$
$$\textit{or} \tag{E.23}$$
$$\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\leq\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)};$$

a model is ***overconfident*** *for the predicted probabilities at position x of a validation image I_i if*

$$-\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)\leq-\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$
$$\textit{or} \tag{E.24}$$
$$\boldsymbol{z}_i(x)^{(S_i(x))}\leq\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}.$$

Furthermore, for underconfidence of semantic segmentation, the NLL is generally less than or equal to the entropy. This is because, when training is insufficient, for correct predictions we have NLL less than or equal to the entropy while for incorrect predictions there is no guaranteed relationship between NLL and entropy. Besides, the majority of the pixel/voxel label predictions for a semantic segmentation are correct after the network has been trained a certain period of time (before overconfidence). Hence, NLL will is expected to be less than or equal to the entropy on average during the underconfident stage. Thus we have the following constraints during underconfidence,

$$\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\geq\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\mathbf{z}_i(x)\right)^{(l)} \tag{E.25}$$
$$\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\geq\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\mathbf{z}_i(x)\right)^{(l)},\quad\forall i \tag{E.26}$$
$$\mathbf{z}_i(x)^{(S_i(x))}\geq\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\mathbf{z}_i(x)\right)^{(l)},\quad\forall i,x. \tag{E.27}$$

Eq. (E.25), Eq. (E.26), and Eq. (E.27) are the prototypes of the constraints for TS, IBTS, LTS in Theorem 3.

Definition 5. *For semantic segmentation, a model is* ***underconfident*** *for the predicted probabilities in n validation images if*

$$-\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)\geq-\sum_{i=1}^{n}\sum_{x\in\Omega}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$

$$\textit{or} \tag{E.28}$$

$$\sum_{i=1}^{n}\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\geq\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)};$$

*a model is **underconfident** for the predicted probabilities in a validation image I_i if*

$$-\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)\geq-\sum_{x\in\Omega}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$

$$\textit{or} \tag{E.29}$$

$$\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\geq\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)};$$

*a model is **underconfident** for the predicted probabilities at position x of a validation image I_i if*

$$-\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)\geq-\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$

$$\textit{or} \tag{E.30}$$

$$\boldsymbol{z}_i(x)^{(S_i(x))}\geq\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}.$$

Definition 6. *For semantic segmentation, a model is **balanced** for the predicted probabilities in n validation images if*

$$-\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)=-\sum_{i=1}^{n}\sum_{x\in\Omega}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$

$$\textit{or} \tag{E.31}$$

$$\sum_{i=1}^{n}\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}=\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)};$$

*a model is **balanced** for the predicted probabilities in a validation image I_i if*

$$-\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)=-\sum_{x\in\Omega}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$

$$\textit{or} \tag{E.32}$$

$$\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}=\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)};$$

*a model is **balanced** for the predicted probabilities at position x of a validation image I_i if*

$$-\sum_{l=1}^{L}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}\right)=-\log\left(\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(S_i(x))}\right)$$

$$\textit{or} \tag{E.33}$$

$$\boldsymbol{z}_i(x)^{(S_i(x))}=\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\left(\boldsymbol{z}_i(x)\right)^{(l)}.$$

E.3. Weighted Averaged Logits and Entropy Extremes

Lemma 2. *Given n logit vector maps $\boldsymbol{z}_1, ..., \boldsymbol{z}_n$, equal probability for all labels is the unique solution q (probability distribution) to the following entropy maximization problem:*

$$\begin{aligned} \max_{q} \quad & -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} q\big(\boldsymbol{z}_i(x)\big)^{(l)} \log\Big(q\big(\boldsymbol{z}_i(x)\big)^{(l)}\Big) \\ \textit{subject to} \quad & q\big(\boldsymbol{z}_i(x)\big)^{(l)} \geq 0 \quad \forall i, x, l \\ & \sum_{l=1}^{L} q\big(\boldsymbol{z}_i(x)\big)^{(l)} = 1 \quad \forall i, x \end{aligned} \tag{E.34}$$

Proof. We use Lagrangian multipliers to solve the optimization problem. $q\big(\mathbf{z}_i(x)\big)^{(l)} \geq 0$ is ignored in the Lagrangian but the deducted solution satisfies this constraint automatically.
Let $\beta_i(x)$ be the multipliers. The Lagrangian is

$$\mathcal{L} = -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} \log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) + \sum_{i=1}^{n}\sum_{x\in\Omega}\beta_i(x)\Big(\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} - 1\Big). \tag{E.35}$$

We take the derivative with respect to $q\big(\mathbf{z}_i(x)\big)^{(l)}$ and set it to 0

$$\frac{\partial \mathcal{L}}{\partial q\big(\mathbf{z}_i(x)\big)^{(l)}} = -1 - \log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) + \beta_i(x) = 0. \tag{E.36}$$

Thus, we obtain the expression of $q\big(\mathbf{z}_i(x)\big)^{(l)}$ as

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = e^{\beta_i(x)-1}. \tag{E.37}$$

Hence, $q\big(\mathbf{z}_i(x)\big)^{(l)} \geq 0$. Since $\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} = 1$ for all i and x, it must satisfy

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = \frac{1}{L}. \tag{E.38}$$

Hence the equal probability distribution over all labels is the entropy maximization solution.

□

Remark. For a classification or semantic segmentation task, equal probability for each label will yield the maximum entropy.

Remark. The minimum entropy lies at extreme points, i.e.

$$\left.\begin{aligned} \arg\min_{q} \quad & -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} \log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) \\ \textit{subject to} \quad & q\big(\mathbf{z}_i(x)\big)^{(l)} \geq 0 \quad \forall i, x, l \\ & \sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} = 1 \quad \forall i, x \end{aligned}\right\} \Big\{q\big(\mathbf{z}_i(x)\big)^{(l)} = 1, q\big(\mathbf{z}_i(x)\big)^{(j)} = 0, (\forall j \neq i)\Big\}, \forall i \tag{E.39}$$

E.4. Entropy Extremes Under Constraints

Theorem 2. *Given n logit vector maps $\boldsymbol{z}_1, ..., \boldsymbol{z}_n$ and label maps $S_1, ..., S_n$, temperature scaling (TS), image-based temperature scaling (IBTS) and local temperature scaling (LTS) are the unique solutions q (probability distribution) to the following entropy maximization problem with different constraints (A, B or C):*

$$\begin{aligned}
\max_{q} \quad & -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} q\big(\boldsymbol{z}_i(x)\big)^{(l)} \log\Big(q\big(\boldsymbol{z}_i(x)\big)^{(l)}\Big) \\
\textit{subject to} \quad & q\big(\boldsymbol{z}_i(x)\big)^{(l)} \geq 0 \quad \forall i, x, l \\
& \sum_{l=1}^{L} q\big(\boldsymbol{z}_i(x)\big)^{(l)} = 1 \quad \forall i, x \\
& \begin{cases}
\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} q\big(\boldsymbol{z}_i(x)\big)^{(l)} \geq \varepsilon^A & \textit{(A: TS constraint)} \\
\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} q\big(\boldsymbol{z}_i(x)\big)^{(l)} \geq \varepsilon_i^B \quad \forall i & \textit{(B: IBTS constraint)} \\
\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} q\big(\boldsymbol{z}_i(x)\big)^{(l)} \geq \varepsilon_i^C(x) \quad \forall i, x & \textit{(C: LTS constraint)}
\end{cases}
\end{aligned} \tag{E.40}$$

where ε^A, ε_i^B and $\varepsilon_i^C(x)$ are the following constants:

$$\begin{aligned}
\varepsilon^A &= \sum_{i=1}^{n}\sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))}\,, \\
\varepsilon_i^B &= \sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))}\,, \\
\varepsilon_i^C(x) &= \boldsymbol{z}_i(x)^{(S_i(x))}\,.
\end{aligned} \tag{E.41}$$

And the corresponding optimal inverse temperature values for TS, IBTS and LTS are

$$\begin{aligned}
&\begin{cases}
\alpha^* = 0, & \textit{if} \quad \sum_{i=1}^{n}\sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \\
\Big\{\alpha^* > 0 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\big(\alpha^* \boldsymbol{z}_i(x)\big)^{(j)} = \sum_{i=1}^{n}\sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise}
\end{cases} \\
&\begin{cases}
\alpha_i^* = 0, & \textit{if} \quad \sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \\
\Big\{\alpha_i^* > 0 \mid \sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\big(\alpha_i^* \boldsymbol{z}_i(x)\big)^{(j)} = \sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise}
\end{cases} \\
&\begin{cases}
\alpha_i(x)^* = 0, & \textit{if} \quad \boldsymbol{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \\
\Big\{\alpha_i(x)^* > 0 \mid \sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\big(\alpha_i(x)^* \boldsymbol{z}_i(x)\big)^{(j)} = \boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise}
\end{cases}
\end{aligned} \tag{E.42}$$

Proof. We use the Karush–Kuhn–Tucker (KKT) conditions to solve the optimization problems. $q\big(\mathbf{z}_i(x)\big)^{(l)} \geq 0$ is ignored for the KKT conditions as the deducted solution satisfies this constraint automatically (i.e., it is inactive).
For constraint A, let α, $\beta_i(x)$ be the multipliers. The Lagrangian is

$$\begin{aligned}
\mathcal{L} = &-\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} \log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) - \sum_{i=1}^{n}\sum_{x\in\Omega} \beta_i(x)\Big(\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} - 1\Big) \\
&- \alpha\Big(\varepsilon^A - \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)}\Big).
\end{aligned} \tag{E.43}$$

Thus, the KKT conditions are

$$\frac{\partial \mathcal{L}}{\partial q\big(\mathbf{z}_i(x)\big)^{(l)}} = -1 - \log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) + \alpha \mathbf{z}_i(x)^{(l)} - \beta_i(x) = 0 \quad \forall i, l, x, \tag{E.44}$$

$$\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} - 1 = 0 \quad \forall i, x, \tag{E.45}$$

$$\varepsilon^A - \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} \leq 0, \tag{E.46}$$

$$\alpha \geq 0, \tag{E.47}$$

$$\alpha\Big(\varepsilon^A - \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) = 0. \tag{E.48}$$

From Eq. (E.44), we obtain the expression of $q\big(\mathbf{z}_i(x)\big)^{(l)}$ as

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = e^{\alpha \mathbf{z}_i(x)^{(l)} - \beta_i(x) - 1}. \tag{E.49}$$

Hence, $q\big(\mathbf{z}_i(x)\big)^{(l)} \geq 0$. Since $\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} = 1$ (Eq. (E.45)) for all i and x, it must satisfy

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = \frac{e^{\alpha \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha \mathbf{z}_i(x)^{(j)}}}. \tag{E.50}$$

From Eq. (E.46), we have

$$\begin{aligned} \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} &= \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{e^{\alpha \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha \mathbf{z}_i(x)^{(j)}}} \\ &\geq \varepsilon^A \\ &= \sum_{i=1}^{n} \sum_{x \in \Omega} \mathbf{z}_i(x)^{(S_i(x))}. \end{aligned} \tag{E.51}$$

Case 1: If $\sum_{i=1}^{n} \sum_{x \in \Omega} \mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L} \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$, then we have

$$\begin{aligned} \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} &= \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{e^{\alpha \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha \mathbf{z}_i(x)^{(j)}}} \\ &\geq \sum_{i=1}^{n} \sum_{x \in \Omega} \mathbf{z}_i(x)^{(S_i(x))} \\ &> \frac{1}{L} \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}. \end{aligned} \tag{E.52}$$

If $\alpha = 0$, then $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ for all i, l and x. Thus, Eq. (E.46) becomes $\varepsilon^A - \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{1}{L} \leq 0$, which violates the $\sum_{i=1}^{n} \sum_{x \in \Omega} \mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L} \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$ assumption. Hence, $\alpha \neq 0$.
Furthermore, we have

$$\frac{1}{L} \sum_{i=1}^{n} \sum_{x \in \Omega} \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} < \sum_{i=1}^{n} \sum_{x \in \Omega} \mathbf{z}_i(x)^{(S_i(x))} \leq \sum_{i=1}^{n} \sum_{x \in \Omega} \max_{l} \{\mathbf{z}_i(x)^{(l)}\}, \tag{E.53}$$

with Lemma 1 and the intermediate value theorem, there must be a unique strictly positive solution α^* for α such that $\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}q\big(\mathbf{z}_i(x)\big)^{(l)}=\varepsilon^A=\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}$. Thus Eq. (E.47) and Eq. (E.48) both hold.

Case 2: If $\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\le\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}$.
If $\alpha\neq 0$, Eq. (E.48) yields $\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}q\big(\mathbf{z}_i(x)\big)^{(l)}=\varepsilon^A=\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}$. With Lemma 1 and intermediate value theorem, there exists a unique non-positive α. This violates Eq. (E.47) and the $\alpha\neq 0$ assumption. Thus, $\alpha=0$.
Furthermore, when $\alpha=0$, it yields $q\big(\mathbf{z}_i(x)\big)^{(l)}=1/L$ for all i, l and x. Take $q\big(\mathbf{z}_i(x)\big)^{(l)}=1/L$ into Eq. (E.46), the inequality holds. Eq. (E.47) and Eq. (E.48) also hold. From Lemma 2, we know that $q\big(\mathbf{z}_i(x)\big)^{(l)}=1/L$ is the solution for entropy maximization of Eq. (E.34). Since Eq. (E.40) is the subproblem of Eq. (E.34), $q\big(\mathbf{z}_i(x)\big)^{(l)}=1/L$ also reaches the entropy maximization of Eq. (E.40).

Overall, the optimal solution is

$$q\big(\mathbf{z}_i(x)\big)^{(l)}=\frac{e^{\alpha^*\mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L}e^{\alpha^*\mathbf{z}_i(x)^{(j)}}},\tag{E.54}$$

with

$$\begin{cases}\alpha^*=0, & \text{if}\quad \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\le\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\\ \{\alpha^*>0\mid\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\frac{e^{\alpha^*\mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L}e^{\alpha^*\mathbf{z}_i(x)^{(j)}}}=\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\}, & \text{otherwise}\end{cases}\tag{E.55}$$

Let $T=\frac{1}{\alpha^*}$ ($\alpha^*\to 0$ as $T\to+\infty$), then this is the TS solution. Note that T does not depend on i and x, which is the same as the temperature value in Eq. (3.3).

For constraint B, let α_i, $\beta_i(x)$ be the multipliers. Then the Lagrangian is

$$\begin{aligned}\mathcal{L}=&-\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}q\big(\mathbf{z}_i(x)\big)^{(l)}\log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big)-\sum_{i=1}^{n}\sum_{x\in\Omega}\beta_i(x)\Big(\sum_{l=1}^{L}q\big(\mathbf{z}_i(x)\big)^{(l)}-1\Big)\\&-\sum_{i=1}^{n}\alpha_i\Big(\varepsilon_i^B-\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}q\big(\mathbf{z}_i(x)\big)^{(l)}\Big).\end{aligned}\tag{E.56}$$

Thus, the KKT conditions are

$$\frac{\partial\mathcal{L}}{\partial q\big(\mathbf{z}_i(x)\big)^{(l)}}=-1-\log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big)+\alpha_i\mathbf{z}_i(x)^{(l)}-\beta_i(x)=0\quad\forall i,l,x,\tag{E.57}$$

$$\sum_{l=1}^{L}q\big(\mathbf{z}_i(x)\big)^{(l)}-1=0\quad\forall i,x,\tag{E.58}$$

$$\varepsilon_i^B-\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}q\big(\mathbf{z}_i(x)\big)^{(l)}\le 0\quad\forall i,\tag{E.59}$$

$$\alpha_i\ge 0\quad\forall i,\tag{E.60}$$

$$\alpha_i\Big(\varepsilon_i^B-\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}q\big(\mathbf{z}_i(x)\big)^{(l)}\Big)=0\quad\forall i.\tag{E.61}$$

From Eq. (E.57), we obtain the expression of $q\big(\mathbf{z}_i(x)\big)^{(l)}$ as

$$q\big(\mathbf{z}_i(x)\big)^{(l)}=e^{\alpha_i\mathbf{z}_i(x)^{(l)}-\beta_i(x)-1}.\tag{E.62}$$

Hence, $q\big(\mathbf{z}_i(x)\big)^{(l)} \geq 0$. Since $\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} = 1$ (Eq. (E.58)) for all i and x, it must have

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = \frac{e^{\alpha_i \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha_i \mathbf{z}_i(x)^{(j)}}}, \tag{E.63}$$

From Eq. (E.59), we have

$$\begin{aligned} \sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} &= \sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{e^{\alpha_i \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha_i \mathbf{z}_i(x)^{(j)}}} \\ &\geq \varepsilon_i^B \\ &= \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))}. \end{aligned} \tag{E.64}$$

Case 1: If $\sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$, then we have

$$\begin{aligned} \sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} &= \sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{e^{\alpha_i \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha_i \mathbf{z}_i(x)^{(j)}}} \\ &\geq \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} \\ &> \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}. \end{aligned} \tag{E.65}$$

If $\alpha_i = 0$, then $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ for all i, l and x. Thus, Eq. (E.59) becomes $\varepsilon_i^B - \sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{1}{L} \leq 0$, which violates the $\sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$ assumption. Hence, $\alpha_i \neq 0$.
Furthermore, we have

$$\frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} < \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} \leq \sum_{x\in\Omega} \max_l \{\mathbf{z}_i(x)^{(l)}\}, \tag{E.66}$$

with Lemma 1 and the intermediate value theorem, there must be a unique strictly positive solution α_i^* for α_i such that $\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} = \varepsilon_i^B = \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))}$. Thus Eq. (E.60) and Eq. (E.61) both hold.

Case 2: If $\sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} < \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$.
If $\alpha_i \neq 0$, Eq. (E.61) yields $\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} = \varepsilon_i^B = \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))}$. With Lemma 1 and the intermediate value theorem, there exists a unique non-positive α_i. This violates Eq. (E.60) and the $\alpha_i \neq 0$ assumption. Thus, $\alpha_i = 0$.
Furthermore, when $\alpha_i = 0$, it yields $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ for all i, l and x. Take $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ into Eq. (E.59), the inequality holds. Eq. (E.60) and Eq. (E.61) also hold. From Lemma 2, we know that $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ is the solution for entropy maximization of Eq. (E.34). Since Eq. (E.40) is the subproblem of Eq. (E.34), $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ also reaches the entropy maximization of Eq. (E.40).

Overall, the optimal solution is

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = \frac{e^{\alpha_i^* \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha_i^* \mathbf{z}_i(x)^{(j)}}}, \tag{E.67}$$

with

$$\begin{cases} \alpha_i^* = 0, & \text{if} \quad \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \\ \{\alpha_i^* > 0 \mid \sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{e^{\alpha_i^* \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha_i^* \mathbf{z}_i(x)^{(j)}}} = \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))}\}, & \text{otherwise} \end{cases} \tag{E.68}$$

Let $T_i = \frac{1}{\alpha_i^*}$ ($\alpha_i^* \to 0$ as $T_i \to +\infty$), then this is the IBTS solution. Note that T_i does not depend on x, which is the same as the temperature value in Eq. (3.4).

For constraint C, let $\alpha_i(x)$, $\beta_i(x)$ be the multipliers. Then the Lagrangian is

$$\begin{aligned}\mathcal{L} = &- \sum_{i=1}^{n} \sum_{x\in\Omega} \sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} \log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) - \sum_{i=1}^{n} \sum_{x\in\Omega} \beta_i(x)\Big(\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} - 1\Big) \\ &- \sum_{i=1}^{n} \sum_{x\in\Omega} \alpha_i(x)\Big(\varepsilon_i^C(x) - \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)}\Big).\end{aligned} \tag{E.69}$$

Thus, the KKT conditions are

$$\frac{\partial \mathcal{L}}{\partial q\big(\mathbf{z}_i(x)\big)^{(l)}} = -1 - \log\Big(q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) + \alpha_i(x)\mathbf{z}_i(x)^{(l)} - \beta_i(x) = 0 \quad \forall i, x, l, \tag{E.70}$$

$$\sum_{l=1}^{L} q\big(\mathbf{z}_i(x)\big)^{(l)} - 1 = 0 \quad \forall i, x, \tag{E.71}$$

$$\varepsilon_i^C(x) - \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} \le 0 \quad \forall i, x, \tag{E.72}$$

$$\alpha_i(x) \ge 0 \quad \forall i, x, \tag{E.73}$$

$$\alpha_i(x)\Big(\varepsilon_i^C(x) - \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)}\Big) = 0 \quad \forall i, x. \tag{E.74}$$

From Eq. (E.70), we obtain the expression of $q\big(\mathbf{z}_i(x)\big)^{(l)}$ as

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = e^{\alpha_i(x)\mathbf{z}_i(x)^{(l)} - \beta_i(x) - 1}. \tag{E.75}$$

Hence, $q\big(\mathbf{z}_i(x)\big)^{(l)} \ge 0$. Since $\sum_{l=1}^{L} q(\mathbf{z}_i(x))^{(l)} = 1$ (Eq. (E.71)) for all i and x, it must have

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = \frac{e^{\alpha_i(x)\mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha_i(x)\mathbf{z}_i(x)^{(j)}}}, \tag{E.76}$$

From Eq. (E.72), we have

$$\begin{aligned}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} &= \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{e^{\alpha_i(x)\mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha_i(x)\mathbf{z}_i(x)^{(j)}}} \\ &\ge \varepsilon_i^C(x) \\ &= \mathbf{z}_i(x)^{(S_i(x))}.\end{aligned} \tag{E.77}$$

Case 1: If $\mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}$, then we have

$$\begin{aligned}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} &= \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \frac{e^{\alpha_i(x)\mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L} e^{\alpha_i(x)\mathbf{z}_i(x)^{(j)}}} \\ &\ge \mathbf{z}_i(x)^{(S_i(x))} \\ &> \frac{1}{L}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)}.\end{aligned} \tag{E.78}$$

If $\alpha_i(x) = 0$, then $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ for all i, l and x. Thus, Eq. (E.72) becomes $\varepsilon_i^C(x) - \sum_{l=1}^L \mathbf{z}_i(x)^{(l)} \frac{1}{L} \leq 0$, which violates the $\mathbf{z}_i(x)^{(S_i(x))} > \frac{1}{L}\sum_{l=1}^L \mathbf{z}_i(x)^{(l)}$ assumption. Hence, $\alpha_i(x) \neq 0$.
Furthermore, we have

$$\frac{1}{L}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} < \mathbf{z}_i(x)^{(S_i(x))} \leq \max_l \{\mathbf{z}_i(x)^{(l)}\}, \tag{E.79}$$

with Lemma 1 and the intermediate value theorem, there must be a unique strictly positive solution $\alpha_i(x)^*$ for $\alpha_i(x)$ such that $\sum_{l=1}^L \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} = \varepsilon_i^C(x) = \mathbf{z}_i(x)^{(S_i(x))}$. Thus Eq. (E.73) and Eq. (E.74) both hold.

Case 2: If $\mathbf{z}_i(x)^{(S_i(x))} < \frac{1}{L}\sum_{l=1}^L \mathbf{z}_i(x)^{(l)}$.
If $\alpha_i(x) \neq 0$, Eq. (E.74) yields $\sum_{l=1}^L \mathbf{z}_i(x)^{(l)} q\big(\mathbf{z}_i(x)\big)^{(l)} = \varepsilon_i^C(x) = \mathbf{z}_i(x)^{(S_i(x))}$. With Lemma 1 and the intermediate value theorem, there exists a unique non-positive α_i. This violates Eq. (E.73) and $\alpha_i(x) \neq 0$ assumption. Thus, $\alpha_i(x) = 0$.
Furthermore, when $\alpha_i(x) = 0$, it yields $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ for all i, l and x. Take $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ into Eq. (E.72), the inequality holds. Eq. (E.73) and Eq. (E.74) also hold. From Lemma 2, we know that $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ is the solution for entropy maximization of Eq. (E.34). Since Eq. (E.40) is the subproblem of Eq. (E.34), $q\big(\mathbf{z}_i(x)\big)^{(l)} = 1/L$ also reaches the entropy maximization of Eq. (E.40).

Overall, the optimal solution is

$$q\big(\mathbf{z}_i(x)\big)^{(l)} = \frac{e^{\alpha_i(x)^* \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^L e^{\alpha_i(x)^* \mathbf{z}_i(x)^{(j)}}}, \tag{E.80}$$

with

$$\begin{cases} \alpha_i(x)^* = 0, & \text{if} \quad \mathbf{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{l=1}^L \mathbf{z}_i(x)^{(l)} \\ \{\alpha_i(x)^* > 0 \mid \sum_{l=1}^L \mathbf{z}_i(x)^{(l)} \frac{e^{\alpha_i(x)^* \mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^L e^{\alpha_i(x)^* \mathbf{z}_i(x)^{(j)}}} = \mathbf{z}_i(x)^{(S_i(x))}\}, & \text{otherwise} \end{cases} \tag{E.81}$$

Let $T_i(x) = \frac{1}{\alpha_i(x)^*}$ ($\alpha_i(x)^* \to 0$ as $T_i(x) \to +\infty$), then this is the LTS solution. Note that this $T_i(x)$ depends on i and x, which is the same as the temperature value in Eq. (3.6).

□

Remark. Note that the first two constraints on $q(\mathrm{z}_i(x))$ are shared by all three models, while the last constraint varies across the three models, i.e. A for TS, B for IBTS, and C for LTS. The first two constraints guarantee that q is a probability distribution while the last constraint makes assumptions on the distributions of the corresponding models. Constraint A assumes that the average true class logit is less than or equal to the weighted average logit over the entire image space and all samples. Constraint B requires that the avearge true class logit is less than or equal to the weighted average logit over the image space. Constraint C specifies that the true class logit is less than or equal to the weighted average logit at each location of each image. Note that the three constrains are designed under the overconfidence scenario. The order of the restrictiveness of the constraints is C > B > A, which indicates the model complexity order LTS > IBTS > TS.

Remark. Theorem 2 gives a more general proof. However, when it comes to TS, IBTS and LTS, we do not necessarily need such strong conditions. Instead we can use the following simplified theorem 2-b.

Theorem 2-b. *Given n logit vector maps $\mathbf{z}_1, ..., \mathbf{z}_n$ and label maps $S_1, ..., S_n$, the optimal temperature values of temperature scaling (TS), image-based temperature scaling (IBTS) and local temperature scaling (LTS) to the following entropy*

maximization problem with different constraints (A, B or C)

$$
\begin{aligned}
\max_{\alpha_i(x)} \quad & -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha_i(x)\boldsymbol{z}_i(x)\big)^{(l)}\log\Big(\sigma_{SM}\big(\alpha_i(x)\boldsymbol{z}_i(x)\big)^{(l)}\Big)\\
\textit{subject to} \quad & \alpha_i(x)\geq 0 \quad \forall i,x,l\\
& \begin{cases}
\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha_i(x)\boldsymbol{z}_i(x)\big)^{(l)}\geq\varepsilon^{A} & \textit{(A: TS constraint)}\\
\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha_i(x)\boldsymbol{z}_i(x)\big)^{(l)}\geq\varepsilon_i^{B} \quad \forall i & \textit{(B: IBTS constraint)}\\
\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha_i(x)\boldsymbol{z}_i(x)\big)^{(l)}\geq\varepsilon_i^{C}(x) \quad \forall i,x & \textit{(C: LTS constraint)}
\end{cases}
\end{aligned}
\tag{E.82}
$$

where ε^{A}, ε_i^{B} *and* $\varepsilon_i^{C}(x)$ *are the following constants:*

$$
\begin{aligned}
\varepsilon^{A} &= \sum_{i=1}^{n}\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))},\\
\varepsilon_i^{B} &= \sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))},\\
\varepsilon_i^{C}(x) &= \boldsymbol{z}_i(x)^{(S_i(x))}.
\end{aligned}
\tag{E.83}
$$

are

$$
\begin{aligned}
&\begin{cases}
\alpha^{*}=0, & \textit{if}\quad \sum_{i=1}^{n}\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\leq\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\\
\Big\{\alpha^{*}>0 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha^{*}\boldsymbol{z}_i(x)\big)^{(j)}=\sum_{i=1}^{n}\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise}
\end{cases}\\
&\begin{cases}
\alpha_i^{*}=0, & \textit{if}\quad \sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\leq\frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\\
\Big\{\alpha_i^{*}>0 \mid \sum_{x\in\Omega}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha_i^{*}\boldsymbol{z}_i(x)\big)^{(j)}=\sum_{x\in\Omega}\boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise}
\end{cases}\\
&\begin{cases}
\alpha_i(x)^{*}=0, & \textit{if}\quad \boldsymbol{z}_i(x)^{(S_i(x))}\leq\frac{1}{L}\sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\\
\Big\{\alpha_i(x)^{*}>0 \mid \sum_{l=1}^{L}\boldsymbol{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha_i(x)^{*}\boldsymbol{z}_i(x)\big)^{(j)}=\boldsymbol{z}_i(x)^{(S_i(x))}\Big\}, & \textit{otherwise}
\end{cases}.
\end{aligned}
\tag{E.84}
$$

where

$$
\begin{aligned}
&\textit{(TS):} \quad \alpha_i(x)\coloneqq\alpha,\forall i,x, \quad \textit{and} \quad T\coloneqq\frac{1}{\alpha},T\in\mathbb{R}^{+}\\
&\textit{(IBTS):} \quad \alpha_i(x)\coloneqq\alpha_i,\forall x, \quad \textit{and} \quad T_i\coloneqq\frac{1}{\alpha_i},T_i\in\mathbb{R}^{+}\\
&\textit{(LTS):} \quad \alpha_i(x)\coloneqq\alpha_i(x), \quad \textit{and} \quad T_i(x)\coloneqq\frac{1}{\alpha_i(x)},T_i(x)\in\mathbb{R}^{+}.
\end{aligned}
\tag{E.85}
$$

Proof. We use the Karush-Kuhn-Tucker (KKT) conditions to solve the optimization problems. $\alpha\geq 0$ is ignored in the Lagrangian and later be validated w.r.t. the deducted solution. For TS, Let λ be the multiplier, the Lagrangian is

$$
\mathcal{L}=-\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\log\Big(\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big)-\lambda\Big(\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}-\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big).
\tag{E.86}
$$

Taking the derivative w.r.t. α, we have

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \alpha} &= -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big)\log\Big(\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big) \\
&\quad -\sum_{i=1}^{n}\sum_{x\in\Omega}\underbrace{\Big[\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big)\Big]}_{=\sum_{l=1}^{L}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\mathbf{z}_i(x)^{(l)} - \underbrace{\sum_{l=1}^{L}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}}_{=1}\sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(j)} = \mathbf{0}} \\
&\quad +\lambda\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\mathbf{z}_i(x)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big) \qquad \text{(E.87)} \\
&= -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big)\Big(\alpha\mathbf{z}_i(x)^{(l)} - \log\big(\sum_{j=1}^{L}\exp(\alpha\mathbf{z}_i(x)^{(j)})\big)\Big) \\
&\quad +\lambda\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\mathbf{z}_i(x)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big) \qquad \text{(E.88)} \\
&= -\alpha\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big) \\
&\quad +\sum_{i=1}^{n}\sum_{x\in\Omega}\underbrace{\Big[\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big)\Big]}_{=\mathbf{0}}\log\big(\sum_{j=1}^{L}\exp(\alpha\mathbf{z}_i(x)^{(j)})\big) \\
&\quad +\lambda\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\mathbf{z}_i(x)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big) \qquad \text{(E.89)} \\
&= (\lambda-\alpha)\sum_{i=1}^{n}\sum_{x\in\Omega}\Big(\sum_{l=1}^{L}\big(\mathbf{z}_i(x)^{(l)}\big)^2\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)} - \big(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\big)^2\Big). \qquad \text{(E.90)}
\end{aligned}$$

Thus, the KKT conditions are

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \alpha} = (\lambda-\alpha)\sum_{i=1}^{n}\sum_{x\in\Omega}\Big(\sum_{l=1}^{L}\big(\mathbf{z}_i(x)^{(l)}\big)^2\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)} - \big(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\big)^2\Big) = 0 \quad \forall i, x, \qquad \text{(E.91)} \\
\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} - \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)} \le 0, \qquad \text{(E.92)} \\
\lambda \ge 0, \qquad \text{(E.93)} \\
\lambda\Big(\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} - \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big) = 0. \qquad \text{(E.94)}
\end{aligned}$$

By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\sum_{l=1}^{L}\left(\mathbf{z}_i(x)^{(l)}\right)^2\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}-\left(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\right)^2\\
&=\left(\sum_{l=1}^{L}\left(\mathbf{z}_i(x)^{(l)}\right)^2\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\right)\underbrace{\left(\sum_{l=1}^{L}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\right)}_{=1}-\left(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\right)^2 &\text{(E.95)}\\
&\geq\left(\sum_{l=1}^{L}|\mathbf{z}_i(x)^{(l)}|\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\right)^2-\left(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\right)^2 &\text{(E.96)}\\
&\geq 0 &\text{(E.97)}
\end{aligned}$$

Hence, we have $\lambda=\alpha$ in Eq. (E.91).
Case 1: If $\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}>\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}$, then we have

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}&=\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\frac{e^{\alpha\mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L}e^{\alpha\mathbf{z}_i(x)^{(j)}}}\\
&\geq\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\\
&>\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}.
\end{aligned}\tag{E.98}$$

If $\alpha=0$, then $\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}=1/L$ for all i, l and x. Thus, Eq. (E.92) becomes $\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}-\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\frac{1}{L}\leq 0$, which violates the $\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}>\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}$ assumption. Hence, $\alpha\neq 0$.
Furthermore, we have

$$\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}<\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\leq\sum_{i=1}^{n}\sum_{x\in\Omega}\max_{l}\{\mathbf{z}_i(x)^{(l)}\},\tag{E.99}$$

with Lemma 1 and the intermediate value theorem, there must be a unique strictly positive solution α^* for α such that $\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}=\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}$. Thus Eq. (E.93) and Eq. (E.94) both hold.

Case 2: If $\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\leq\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}$.
If $\alpha\neq 0$, Eq. (E.94) and $\lambda=\alpha$ yields $\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}q\left(\mathbf{z}_i(x)\right)^{(l)}=\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}$. With Lemma 1 and the intermediate value theorem, there exists a unique non-positive α. This violates Eq. (E.93) and the $\alpha\neq 0$ assumption. Thus, $\alpha=0$.
Furthermore, when $\alpha=0$, it yields $\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}=1/L$ for all i, l and x. Take $\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}=1/L$ into Eq. (E.92), the inequality holds. Eq. (E.93) and Eq. (E.94) also hold. From Lemma 2, we know that $\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}=1/L$ is the solution for entropy maximization of Eq. (E.34). Since Eq. (E.82) is the subproblem of Eq. (E.34), $\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}=1/L$ also reaches the entropy maximization of Eq. (E.82).

Overall, the optimal solution is

$$\begin{cases}\alpha^*=0, & \text{if}\quad \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\leq\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\\ \{\alpha^*>0\mid\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\frac{e^{\alpha^*\mathbf{z}_i(x)^{(l)}}}{\sum_{j=1}^{L}e^{\alpha^*\mathbf{z}_i(x)^{(j)}}}=\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\}, & \text{otherwise}\end{cases}\tag{E.100}$$

Let $T = \frac{1}{\alpha^*}$ ($\alpha^* \to 0$ as $T \to +\infty$), then this is the TS solution. Note that T does not depend on i and x, which is the same as the temperature value in Eq. (3.3).

Similarly, for IBTS and LTS, we can get

$$\begin{aligned}
&\arg\max_{\alpha_i} -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\alpha_i \mathbf{z}_i(x)\right)^{(l)} \log\left(\sigma_{SM}\left(\alpha_i \mathbf{z}_i(x)\right)^{(l)}\right) \\
=&\begin{cases} \alpha_i^* = 0, & \text{if} \quad \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))} \le \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \\ \left\{\alpha_i^* > 0 \mid \sum_{x\in\Omega}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \sigma_{SM}\left(\alpha_i^* \mathbf{z}_i(x)\right)^{(j)} = \sum_{x\in\Omega} \mathbf{z}_i(x)^{(S_i(x))}\right\}, & \text{otherwise} \end{cases}
\end{aligned} \tag{E.101}$$

$$\begin{aligned}
&\arg\max_{\alpha_i(x)} -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\alpha_i(x) \mathbf{z}_i(x)\right)^{(l)} \log\left(\sigma_{SM}\left(\alpha_i(x) \mathbf{z}_i(x)\right)^{(l)}\right) \\
=&\begin{cases} \alpha_i(x)^* = 0, & \text{if} \quad \mathbf{z}_i(x)^{(S_i(x))} \le \frac{1}{L}\sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \\ \left\{\alpha_i(x)^* > 0 \mid \sum_{l=1}^{L} \mathbf{z}_i(x)^{(l)} \sigma_{SM}\left(\alpha_i(x)^* \mathbf{z}_i(x)\right)^{(j)} = \mathbf{z}_i(x)^{(S_i(x))}\right\}, & \text{otherwise} \end{cases}
\end{aligned} \tag{E.102}$$

□

Theorem 3. *Given n logit vector maps $\boldsymbol{z}_1, ..., \boldsymbol{z}_n$ and label maps $S_1, ..., S_n$, the optimal temperature values of temperature scaling (TS), image-based temperature scaling (IBTS) and local temperature scaling (LTS) to the following entropy minimization problem with different constraints (A, B or C)*

$$\begin{aligned}
\min_{\alpha_i(x)} \quad & -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\alpha_i(x) \boldsymbol{z}_i(x)\right)^{(l)} \log\left(\sigma_{SM}\left(\alpha_i(x) \boldsymbol{z}_i(x)\right)^{(l)}\right) \\
\textit{subject to} \quad & \alpha_i(x) \ge 0 \quad \forall i, x, l \\
& \begin{cases} \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\left(\alpha_i(x) \boldsymbol{z}_i(x)\right)^{(l)} \le \varepsilon^A & \textit{(A: TS constraint)} \\ \sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\left(\alpha_i(x) \boldsymbol{z}_i(x)\right)^{(l)} \le \varepsilon_i^B \quad \forall i & \textit{(B: IBTS constraint)} \\ \sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\left(\alpha_i(x) \boldsymbol{z}_i(x)\right)^{(l)} \le \varepsilon_i^C(x) \quad \forall i, x & \textit{(C: LTS constraint)} \end{cases}
\end{aligned} \tag{E.103}$$

where ε^A, ε_i^B and $\varepsilon_i^C(x)$ are the following constants:

$$\begin{aligned}
\varepsilon^A &= \sum_{i=1}^{n}\sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))} \ge \frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)}, \\
\varepsilon_i^B &= \sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))} \ge \frac{1}{L}\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)}, \\
\varepsilon_i^C(x) &= \boldsymbol{z}_i(x)^{(S_i(x))} \ge \frac{1}{L}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)}.
\end{aligned} \tag{E.104}$$

are

$$\begin{aligned}
&\left\{\alpha^* \ge 0 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\left(\alpha^* \boldsymbol{z}_i(x)\right)^{(j)} = \sum_{i=1}^{n}\sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))}\right\}, \\
&\left\{\alpha_i^* \ge 0 \mid \sum_{x\in\Omega}\sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\left(\alpha_i^* \boldsymbol{z}_i(x)\right)^{(j)} = \sum_{x\in\Omega} \boldsymbol{z}_i(x)^{(S_i(x))}\right\}, \\
&\left\{\alpha_i(x)^* \ge 0 \mid \sum_{l=1}^{L} \boldsymbol{z}_i(x)^{(l)} \sigma_{SM}\left(\alpha_i(x)^* \boldsymbol{z}_i(x)\right)^{(j)} = \boldsymbol{z}_i(x)^{(S_i(x))}\right\}.
\end{aligned} \tag{E.105}$$

where

$$\begin{aligned}
&\textit{(TS):} \quad \alpha_i(x) \coloneqq \alpha, \forall i, x, \quad \textit{and} \quad T \coloneqq \frac{1}{\alpha}, T \in \mathbb{R}^+ \\
&\textit{(IBTS):} \quad \alpha_i(x) \coloneqq \alpha_i, \forall x, \quad \textit{and} \quad T_i \coloneqq \frac{1}{\alpha_i}, T_i \in \mathbb{R}^+ \\
&\textit{(LTS):} \quad \alpha_i(x) \coloneqq \alpha_i(x), \quad \textit{and} \quad T_i(x) \coloneqq \frac{1}{\alpha_i(x)}, T_i(x) \in \mathbb{R}^+.
\end{aligned} \tag{E.106}$$

Proof. For TS, Let

$$\mathcal{F}(\alpha) = -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\log\Big(\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big). \tag{E.107}$$

Taking the derivative w.r.t. α, we have

$$\begin{aligned}
\frac{\partial\mathcal{F}(\alpha)}{\partial\alpha} &= -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big)\log\Big(\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big) \\
&\quad - \sum_{i=1}^{n}\sum_{x\in\Omega}\underbrace{\Big[\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big)\Big]}_{=\sum_{l=1}^{L}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\mathbf{z}_i(x)^{(l)} - \underbrace{\sum_{l=1}^{L}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}}_{=1}\sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(j)} = \mathbf{0}}
\end{aligned} \tag{E.108}$$

$$= -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big)\Big(\alpha\mathbf{z}_i(x)^{(l)} - \log\big(\sum_{j=1}^{L}\exp(\alpha\mathbf{z}_i(x)^{(j)})\big)\Big) \tag{E.109}$$

$$\begin{aligned}
&= -\alpha\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big) \\
&\quad + \sum_{i=1}^{n}\sum_{x\in\Omega}\underbrace{\Big[\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big(\mathbf{z}_i(x)^{(l)} - \sum_{j=1}^{L}\mathbf{z}_i(x)^{(j)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(j)}\Big)\Big]}_{=\mathbf{0}}\log\big(\sum_{j=1}^{L}\exp(\alpha\mathbf{z}_i(x)^{(j)})\big)
\end{aligned} \tag{E.110}$$

$$= -\alpha\sum_{i=1}^{n}\sum_{x\in\Omega}\Big(\sum_{l=1}^{L}\big(\mathbf{z}_i(x)^{(l)}\big)^2\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)} - \big(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\big)^2\Big). \tag{E.111}$$

By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\sum_{l=1}^{L}\big(\mathbf{z}_i(x)^{(l)}\big)^2\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)} - \big(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\big)^2 \\
&= \Big(\sum_{l=1}^{L}\big(\mathbf{z}_i(x)^{(l)}\big)^2\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big)\underbrace{\Big(\sum_{l=1}^{L}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big)}_{=1} - \Big(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big)^2
\end{aligned} \tag{E.112}$$

$$\geq \Big(\sum_{l=1}^{L}|\mathbf{z}_i(x)^{(l)}|\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big)^2 - \Big(\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\big(\alpha\mathbf{z}_i(x)\big)^{(l)}\Big)^2 \tag{E.113}$$

$$\geq 0 \tag{E.114}$$

Since $\alpha \geq 0$, finally we get

$$\frac{\partial \mathcal{F}(\alpha)}{\partial \alpha} \leq 0. \tag{E.115}$$

Thus $\mathcal{F}(\alpha)$ is monotonicly decreasing w.r.t. α.

Furthermore, we have the following relations by definition

$$\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)} \leq \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \tag{E.116}$$

$$\frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)} \leq \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \leq \sum_{i=1}^{n}\sum_{x\in\Omega}\max_{l}\{\mathbf{z}_i(x)^{(l)}\}. \tag{E.117}$$

With Lemma 1 and the intermediate value theorem, there must be a unique non-negative solution α^* for α such that $\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}$. This α^* is also the maximum α that we can get without violating the constraints. Because $\mathcal{F}(\alpha)$ is monotonicly decreasing, thus α^* is the optimal point that minimizes the entropy, i.e.

$$\begin{aligned}
&\arg\min_{\alpha} -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\alpha\mathbf{z}_i(x)\right)^{(l)}\right)\\
=&\left\{\alpha^* \geq 0 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha^*\mathbf{z}_i(x)\right)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\right\}
\end{aligned} \tag{E.118}$$

Similarly, for IBTS and LTS, we can get

$$\begin{aligned}
&\arg\min_{\alpha_i} -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\alpha_i\mathbf{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\alpha_i\mathbf{z}_i(x)\right)^{(l)}\right)\\
=&\left\{\alpha_i^* \geq 0 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha_i^*\mathbf{z}_i(x)\right)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\right\}
\end{aligned} \tag{E.119}$$

$$\begin{aligned}
&\arg\min_{\alpha_i(x)} -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\sigma_{SM}\left(\alpha_i(x)\mathbf{z}_i(x)\right)^{(l)}\log\left(\sigma_{SM}\left(\alpha_i(x)\mathbf{z}_i(x)\right)^{(l)}\right)\\
=&\left\{\alpha_i(x)^* \geq 0 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha_i(x)^*\mathbf{z}_i(x)\right)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\right\}
\end{aligned} \tag{E.120}$$

□

Remark. Different from the proof in Theorem 2 where we used KKT conditions, we only used the gradient here and gave a specific expression for the probability (i.e. softmax of logits) to prove Theorem 3. This kind of proof choice is because (1) the objective function in Theorem 2 is concave and we want to obtain the maximum; (2) the constraints in Theorem 2 are strong enough (self-contained) to derive the solution.

E.5. (Local) Temperature Scaling Drives NLL and Entropy to an Equilibrium

Theorem 4. *(1) When the to-be-calibrated semantic segmentation network is overconfident, minimizing NLL w.r.t. TS, IBTS, and LTS results in solutions that are also the solutions of maximizing entropy of the calibrated probability w.r.t. TS, IBTS and LTS under the condition of overconfidence. (2) When the to-be-calibrated semantic segmentation network is underconfident, minimizing NLL w.r.t. TS, IBTS, and LTS results in solutions that are also the solutions of minimizing entropy of the calibrated probability w.r.t. TS, IBTS and LTS under the condition of underconfidence. (3) The post-hoc probability calibration of semantic segmentation with TS, IBTS and LTS approaches reach an equilibrium between Negative Log Likelihood (NLL) and entropy for both underconfidence and overconfidence.*

Proof. For TS, if overconfident, we have the following relationship from definition 4:

$$\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \leq \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\mathbf{z}_i(x)\right)^{(l)}. \tag{E.121}$$

To eliminate overconfidence, we need to decrease NLL and increase entropy to probabilistically describe empirically observable segmentation errors (see §3.5 for detailed explanations). From Eq. (E.121), Theorem 2 (or theorem 2-b) and Theorem 1 we know there is a unique optimal α^*

$$\begin{cases} \alpha^* = 0, & \text{if} \quad \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)} \\ \left\{0<\alpha^*\leq 1 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha^*\mathbf{z}_i(x)\right)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\right\}, & \text{otherwise} \end{cases} \tag{E.122}$$

that drives the NLL to minimum point and the entropy to maximum point simultaneously. Besides, at the optimal point, NLL equals to entropy, thus reaching an equilibrium. And the overconfidence state is transferred to a balanced state

$$\begin{cases} -\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\frac{1}{L}\log\left(\frac{1}{L}\right) = -\sum_{i=1}^{n}\sum_{x\in\Omega}\log\left(\frac{1}{L}\right), \text{if} \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \leq \frac{1}{L}\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)} \\ \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha^*\mathbf{z}_i(x)\right)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}, \text{otherwise.} \end{cases} \tag{E.123}$$

If underconfident, we have the following relationship from definition 5:

$$\sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \geq \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\mathbf{z}_i(x)\right)^{(l)}. \tag{E.124}$$

To eliminate underconfidence, we need to decrease NLL and decrease entropy to probabilistically describe empirically observable segmentation errors. From Eq. (E.124), Theorem 3 and Theorem 1 we know there is a unique optimal α^*

$$\left\{\alpha^* \geq 1 \mid \sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha^*\mathbf{z}_i(x)\right)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))}\right\} \tag{E.125}$$

that drives the NLL to minimum point and the entropy to minimum point simultanously. Besides, at the optimal point, NLL equals to entropy, thus reaching an equilibrium. And the underconfidence state is transferred to a balanced state

$$\sum_{i=1}^{n}\sum_{x\in\Omega}\sum_{l=1}^{L}\mathbf{z}_i(x)^{(l)}\sigma_{SM}\left(\alpha^*\mathbf{z}_i(x)\right)^{(l)} = \sum_{i=1}^{n}\sum_{x\in\Omega}\mathbf{z}_i(x)^{(S_i(x))} \tag{E.126}$$

Overall, TS post-hoc probability calibration makes NLL and entropy reach an equilibrium for the validation dataset under both the underconfidence and overconfidence scenarios.

Similarly, IBTS and LTS post-hoc probability calibrations also make NLL and entropy reach an equilibrium for each image and for each location respectively under both the underconfident and overconfident scenarios.

□

F. Evaluation Metrics for Semantic Segmentation

This section introduces evaluation metrics for calibration and segmentation.

Reliability Diagram. Reliability diagrams are commonly used as visual representations of calibration performance [11, 53, 56]. A reliability diagram is derived from the definition of perfect calibration where the accuracy and the confidence are presented separately. If a model is perfectly calibrated, then the diagram should indicate an identity relationship between the confidence and the accuracy. Otherwise, there is miscalibration in the model. See Fig. 2 and Fig. 5 for examples.

To visually illustrate the relationship of the confidence and the accuracy in Eq. (3.2), one can estimate both the confidence and the accuracy from finite samples. Specifically, semantic segmentation results can be grouped into N equal-sized probability intervals (each of size $1/N$) to calculate the accuracy of each bin. Let Ω_j be the set of pixels/voxels whose predicted probabilities fall into the interval $\Delta_j = (\frac{j-1}{N}, \frac{j}{N}]$. Thus, the *accuracy* [20] of Ω_j can be estimated as

$$acc(\Omega_j) = \frac{1}{|\Omega_j|} \sum_{x \in \Omega_j} \mathbb{1}(\hat{S}(x) = S(x)), \tag{F.1}$$

where $\hat{S}(x)$ and $S(x)$ are the predicted and true labels for pixel/voxel x, $\mathbb{1}$ is the indicator function. Note that $acc(\Omega_j)$ is an unbiased and consistent estimator of $\mathbb{P}(\hat{S} = S|\hat{P} \in \Delta_j)$ [20] where $\hat{P}(x)$ is the probability associated with $\hat{S}(x)$ for pixel/voxel at location x. The *average confidence* [20] over bin Ω_j can be defined as

$$conf(\Omega_j) = \frac{1}{|\Omega_j|} \sum_{x \in \Omega_j} \hat{P}(x), \tag{F.2}$$

Thus, $acc(\Omega_j)$ and $conf(\Omega_j)$ approximate the left-hand side and right-hand side of Eq. (3.2) for bin Ω_j.

Based on the definition of perfect calibration, a reliability diagram checks whether $acc(\Omega_j) = conf(\Omega_j)$ for all $j \in 1, 2, ..., N$ and plots the quantitative relation in a bar chart.

Expected Calibration Error (ECE). A reliability diagram is only a visual cue to indicate the performance of model calibration: it does not reflect the number of pixels/voxels in each bin. Thus, to account for such variations of the number of samples in a bin, it has been suggested [54] to use a scalar value to summarize the overall calibration performance. The expected calibration error [54] uses the expectation between confidence and the accuracy to indicate the magnitude of the miscalibration. More precisely,

$$ECE = \sum_{j=1}^{N} \frac{|\Omega_j|}{\Omega_*} |acc(\Omega_j) - conf(\Omega_j)|, \tag{F.3}$$

where $\Omega_* = \sum_j^N |\Omega_j|$ is the total number of pixels/voxels. The difference between acc and $conf$ for a given bin represents the calibration gap.

Maximum Calibration Error (MCE). The maximum calibration error [54] measures the worst-case deviation between the confidence and the accuracy. This is extremely important in high-risk applications where reliable confidence prediction is crucial for decision making. Specifically,

$$MCE = \max_{j \in \{1,...,N\}} |acc(\Omega_j) - conf(\Omega_j)|. \tag{F.4}$$

Note that both the ECE and the MCE are closely related to the reliability diagram. The ECE is a weighted average of all gaps across all bins while the MCE is the largest gap.

Static Calibration Error (SCE). The ECE is computed by only using the predicted label's probability, which does not consider information obtained for other labels. The static calibration error (SCE) [57] has therefore been proposed for the multi-label setting, which extends ECE by separately computing the calibration error within a bin for each label followed by averaging across all bins. More precisely, the SCE is defined as

$$SCE = \sum_{l \in L} \sum_{j=1}^{N} \frac{|\Omega_{j,l}|}{|L|\Omega_*} |acc(\Omega_{j,l}) - conf(\Omega_{j,l})|, \tag{F.5}$$

where L is the set of labels, $\Omega_{j,l}$ is the subset of pixels/voxels for label l in bin Ω_j.

Adaptive Calibration Error (ACE). Another weakness of ECE is that the number of pixels/voxels in each bin varies a lot among different bins, posing a bias-variance tradeoff for choosing the number of bins [57]. This motivates the introduction

of the adaptive calibration error (ACE) [57]. Specifically, ACE uses an adaptive scheme which separates the bin intervals so that each bin contains an equal number of pixels/voxels. Specifically,

$$ACE = \sum_{l \in L} \sum_{r=1}^{R} \frac{1}{|L|R} |acc(\Omega_{r,l}) - conf(\Omega_{r,l})|, \tag{F.6}$$

where R is the number of equal-frequency bins, Ω_r is the r-th sorted bin which contains Ω_*/R pixels/voxels. $\Omega_{r,l}$ is the subset of pixels/voxels for label l in the r-th bin Ω_r.

Avgerage Surface Distance (ASD). ASD is the symmetric average surface distance (usually in millimeter (mm)) between each predicted segmentation label and the true segmentation label. The distance between a point p on a gold-standard or ground-truth surface $\partial S^{(l)}$ and the predicted surface $\partial \hat{S}^{(l)}$ with respect to label l is given by the minimum of the Euclidean norm, i.e. $d(p, \partial \hat{S}^{(l)}) = \min_{\hat{p} \in \partial \hat{S}^{(l)}} ||p - \hat{p}||_2$, where $\hat{p}$ is a point on surface $\partial \hat{S}^{(l)}$. Hence symmetric average surface distance is defined as

$$ASD = \frac{1}{|L|} \sum_{l \in L} \left(\frac{1}{|\partial S^{(l)}| + |\partial \hat{S}^{(l)}|} \left(\sum_{p \in \partial S^{(l)}} d(p, \partial \hat{S}^{(l)}) + \sum_{\hat{p} \in \partial \hat{S}^{(l)}} d(\hat{p}, \partial S^{(l)}) \right) \right). \tag{F.7}$$

Surface Dice (SD). SD is the averaged Dice score between the segmented label surface and the true label surface at a given tolerance (we use 1 mm). This tolerance captures that a point p may still be counted as being on the surface $\partial \hat{S}^{(l)}$ if the distance is at or below the tolerance, i.e. $d(p, \partial \hat{S}^{(l)}) \leq$ tolerance. Formally, the averaged surface Dice score is defined as

$$SD = \frac{1}{|L|} \sum_{l \in L} \frac{2|\{p | d(p, \partial S^{(l)}) \leq \epsilon, d(p, \partial \hat{S}^{(l)}) \leq \epsilon\}|}{|\{p | d(p, \partial S^{(l)}) \leq \epsilon\}| + |\{p | d(p, \partial \hat{S}^{(l)}) \leq \epsilon\}|}, \tag{F.8}$$

where ϵ is the tolerance threshold, and $|\cdot|$ is the Cardinality of the set.

95% Maximum Distance (95MD). 95MD is the 95th percentile of the symmetric distance between the segmented label volume and the true label volume. The definition is

$$95MD = \frac{1}{|L|} \sum_{l \in L} \left(95\%\text{Percentile} \left\{ ..., d(p, \hat{S}^{(l)}), ..., d(\hat{p}, S^{(l)}), ... \right\} \quad \forall p \in S^{(l)}, \hat{p} \in \hat{S}^{(l)} \right). \tag{F.9}$$

Volume Dice (VD). VD is the average Dice score over segmented labels (excluding the background). This is a commonly used metric to determine the success of segmentation in the field of medical image analysis. It is defined as

$$VD = \frac{1}{|L|} \sum_{l \in L} \frac{2|S^{(l)} \cap \hat{S}^{(l)}|}{|S^{(l)}| + |\hat{S}^{(l)}|}. \tag{F.10}$$

G. Example of *Boundary* Region and *All* Region

Fig. 6 shows an example of the *Boundary* region and the *All* region for a 2D slice of a 3D MR brain image. The *Boundary* region is created with boundaries of labels and voxels that are up to 2 voxels away from boundary voxels. The *All* region contains label regions excluding the background and the *Boundary* region. Note that in the multi-atlas segmentation label fusion experiment, the boundary region of the VoteNet+ ground-truth labels is very sparse and thin. Thus, we use the *Boundary* region and the *All* region of the original segmentation labels of the magnetic resonance (MR) images instead. This is the same evaluation approach as for the U-Net segmentation experiment.

H. Patch Size vs Metrics Results

Fig. 7 shows the results of *Local-Avg* for different metrics with different patch sizes. Note that the *Local-Avg* and *Local-Max* results reported in Tab. 1 are for a patch size of 72×72 (or 72×72×72 in 3D). We observe that the probability calibration performance tends to be worse for smaller patch sizes. This is expected as patch variations (also the differences of patch-based multi-class probability distributions) are very significant across patches when patch sizes are small. LTS can improve the calibration performance over TS and IBTS, because it can capture spatially varying effects.

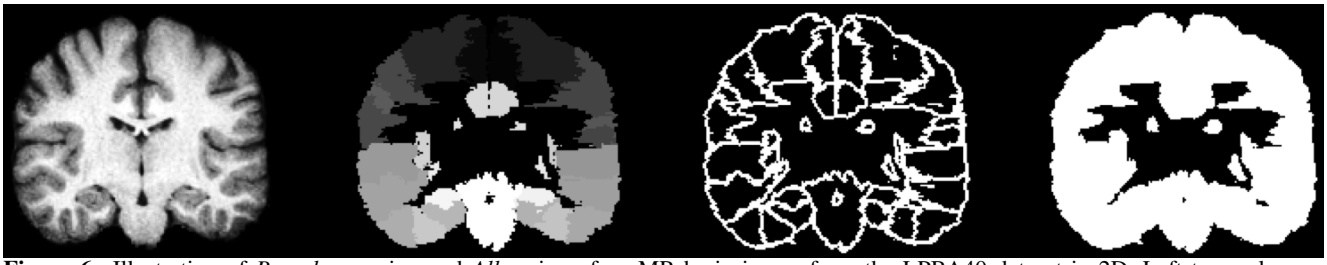

Figure 6: Illustration of *Boundary* region and *All* region of an MR brain image from the LPBA40 dataset in 2D. Left two columns: image and corresponding label map. Right two columns: *Boundary* region and *All* region. The *Boundary* region is usually where mis-segmentations and mis-calibrations occur. The *All* region enlarges the label region to include the *Boundray* region, it thus captures an evaluation region which excludes almost all background of an image.

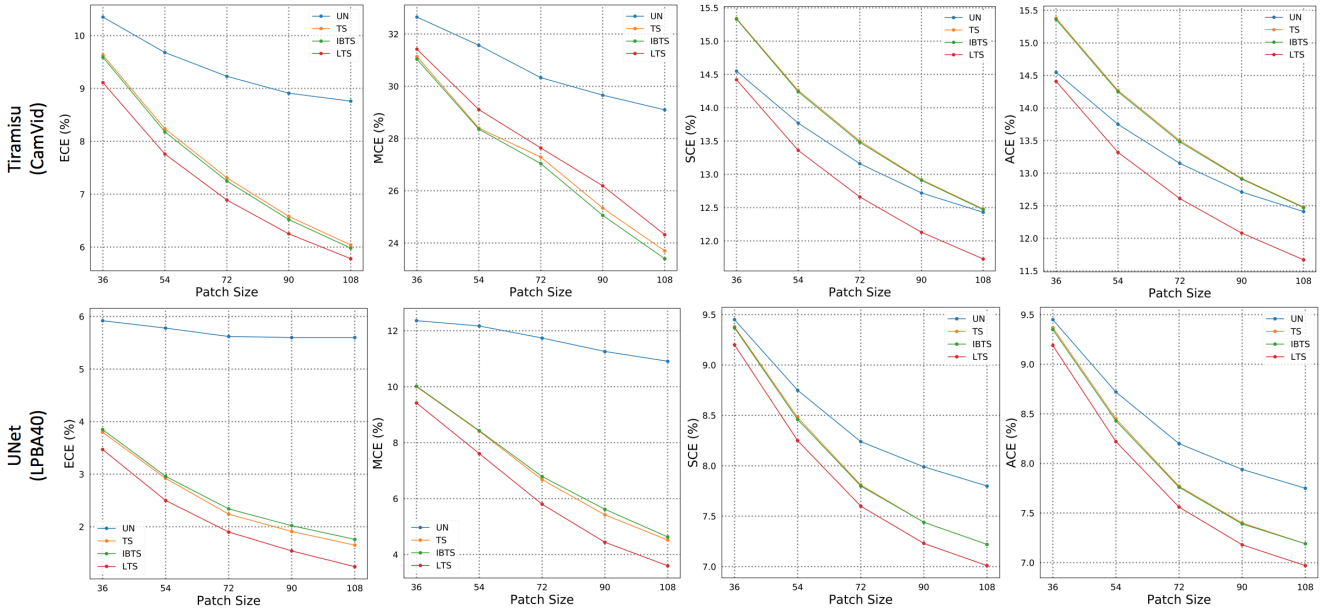

Figure 7: *Local-Avg* results LPBA40 and CamVid experiments for different patch sizes. UN denotes uncalibrated results. In general, the smaller the patch size the worse the performance. Besides, LTS works best for most metrics.

I. Dataset Variations

Image variations are different for different datasets. Fig. 8 illustrates such variations. COCO using an FCN is the most complex dataset, followed by CamVid using Tiramisu, LPBA40 using a UNet and finally LPBA40 combined with VoteNet+. The quantitative results of the metrics in Tab. 1 follows the same pattern: with the results for COCO using an FCN the weakest and the results for LPBA40 using VoteNet+ the best.

J. Additional Quantitative Results

Additional quantitative results are provided in Tab. 3. The results are in line with the conclusions we obtain in §4, i.e. LTS works significantly better than TS [20], isotonic Regression (IsoReg) [68], ensemble temperature scaling (ETS) [69], vector scaling (VS) [20], and Dirichlet calibration with off-diagonal regularization (DirODIR) [34].

K. Multi-atlas Segmentation and Joint Label Fusion

We give a brief overview of multi-atlas segmentation (MAS) [26] and label fusion. Let T_I represents the target image that needs to be segmented. Denote the n atlas images and their corresponding manual segmentations as $A^1 = (A^i_I, A^i_S), A^2 = (A^2_I, A^2_S), ..., A^n = (A^n_I, A^n_S)$. MAS first employs a reliable deformable image registration method to warp all atlas images into the space of the target image T_I, i.e. $\tilde{A}^i = (\tilde{A}^i_I, \tilde{A}^i_S), i = 1, ..., n$. Each $\tilde{A}^i_S$ is considered as a candidate segmentation

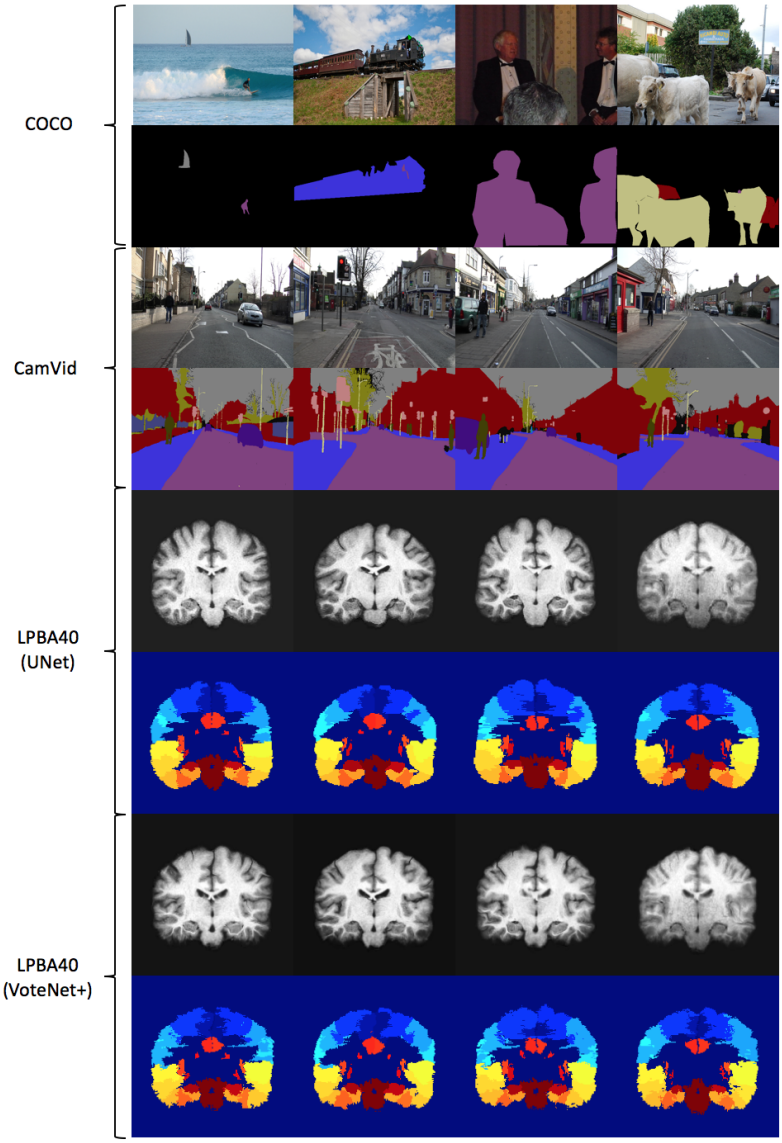

Figure 8: An example of images and labels in different datasets for different experiments. COCO is the most complex dataset and contains different kinds of natural images. CamVid is mainly focused on street scenes. LPBA40 is a dataset of 3D brain MR images. Note that images for UNet are affine pre-registered to a common atlas space while images for VoteNet+ are registered to a target image via a deformable registration. Thus image variations of VoteNet+ experiment are less than that for the UNet experiment.

for T_I. Finally, a label fusion method [26] $\mathscr{G}$ is used to produce the final estimated segmentation $\hat{T}_S$ for T_I, i.e.

$$\hat{T}_S = \mathscr{G}(\tilde{A}^1, \tilde{A}^2, ..., \tilde{A}^n, T_I). \tag{K.1}$$

The goal of label fusion is to use all the information from each individual candidate segmentation to generate a consensus segmentation that is better than any individual candidate segmentation. One of the most common and popular approaches of label fusion is weighted voting at each pixel/voxel of the target image, i.e.

$$\hat{T}_S(x) = \arg\max_{l \in L} \sum_{i=1}^{n} w_x^i \cdot \mathbb{1}[\tilde{A}_S^i(x) = l], \tag{K.2}$$

where $l \in L = \{0, \ldots, K\}$ is the set of labels (K structures; 0 indicating background), $\mathbb{1}[\cdot]$ is the indicator function, and w_x^i is the weight that associates with the i-th atlas candidate segmentation $\tilde{A}_S^i$ at position x. There are a lot of possible

Dataset	Method	ECE(%)↓			MCE(%)↓			SCE(%)↓			ACE(%)↓		
		All	*Boundary*	*Local-Avg* [*Local-Max*]	*All*	*Boundary*	*Local-Avg* [*Local-Max*]	*All*	*Boundary*	*Local-Avg* [*Local-Max*]	*All*	*Boundary*	*Local-Avg* [*Local-Max*]
Tiramisu CamVid (233)	UC	7.79(4.94)	22.79(5.76)	9.23(10.63) [25.35(12.80)]	22.64(12.72)	30.42(10.65)	30.33(16.63) [56.15(14.61)]	9.91(5.02)	24.62(5.69)	13.16(11.72) [30.60(12.48)]	9.90(5.01)	24.43(5.75)	13.15(11.73) [30.60(12.46)]
	IsoReg [68]	3.77(3.71)	16.86(5.99)	7.79(8.56) [21.18(12.73)]	18.19(11.70)	24.59(10.00)	27.66(15.89) [40.66(20.14)]	9.91(3.86)	19.89(5.65)	13.94(10.71) [29.79(12.51)]	10.07(3.85)	19.72(5.70)	14.08(10.74) [29.92(12.45)]
	VS [20]	5.85(4.27)	17.95(6.46)	11.24(11.11) [24.97(14.50)]	21.14(8.44)	32.25(12.68)	38.47(18.10) [44.92(19.20)]	10.84(5.56)	22.84(5.62)	14.90(12.59) [31.13(14.99)]	10.80(5.55)	22.39(5.73)	14.83(12.62) [31.01(14.95)]
	ETS [69]	3.71(3.65)	16.28(6.08)	7.76(8.46) [20.86(12.73)]	17.63(10.33)	23.06(9.25)	27.63(15.94) [41.09(20.13)]	9.98(3.85)	19.48(5.62)	14.05(10.70) [29.78(12.46)]	10.12(3.84)	19.30(5.67)	14.14(10.72) [29.85(12.42)]
	DirODIR [34]	6.63(5.51)	25.32(8.14)	11.79(13.66) [25.01(16.57)]	15.77(8.27)	34.92(11.45)	33.54(19.77) [43.56(22.37)]	12.42(7.33)	29.01(7.26)	17.33(16.00) [32.75(18.49)]	12.37(7.34)	28.84(7.33)	17.32(16.00) [32.66(18.42)]
	TS [20]	3.45(3.52)	12.66(5.43)	7.31(7.72) [17.69(11.91)]	16.02(11.09)	23.57(12.88)	27.29(16.23) [37.25(18.98)]	9.42(3.90)	17.85(4.55)	13.50(10.14) [27.72(11.37)]	9.44(3.92)	17.61(4.59)	13.50(10.17) [27.76(11.33)]
	IBTS	3.63(3.65)	12.57(6.07)	7.25(7.67) [17.60(11.91)]	16.01(10.21)	23.24(13.00)	27.04(15.94) [37.61(19.27)]	9.47(3.89)	17.98(4.88)	13.48(10.12) [27.69(11.38)]	9.49(3.91)	17.75(4.92)	13.48(10.16) [27.76(11.33)]
	LTS	3.40(3.59)	11.80(5.20)	**6.89(7.64)** [16.61(11.81)]	**12.44(7.48)**	22.17(9.53)	27.64(16.67) [37.92(20.47)]	8.76(4.05)	17.77(4.26)	12.66(10.04) [26.78(11.22)]	8.73(4.03)	17.32(4.32)	12.61(10.07) [26.76(11.22)]
	MMCE [36]	4.45(4.03)	–	– [–]	18.83(10.82)	–	– [–]	8.59(5.98)	–	– [–]	8.50(5.00)	–	– [–]
	MMCE [36]+**LTS**	4.15(3.54)	–	– [–]	17.98(10.69)	–	– [–]	7.28(3.80)	–	– [–]	7.17(3.84)	–	– [–]
	FL [52]	3.47(3.11)	**8.68(5.45)**	9.01(7.19) [13.84(11.67)]	14.77(13.28)	**17.62(13.53)**	28.37(15.86) [33.33(18.08)]	7.46(3.43)	**14.08(4.49)**	14.09(9.78) [23.60(12.11)]	7.43(3.45)	**13.63(4.57)**	14.06(9.83) [23.62(12.05)]
	FL [52]+**LTS**	**3.13(3.64)**	11.06(5.55)	6.96(8.21) [**12.66(12.87)**]	14.51(11.07)	19.61(9.82)	**26.91(16.06)** [**32.27(19.08)**]	**6.78(4.05)**	15.28(4.76)	**11.85(10.69)** [**22.04(13.05)**]	**6.73(4.05)**	14.76(4.84)	**11.83(10.73)** [**22.10(12.96)**]

Table 3: Calibration results for Tiramisu semantic segmentation model on CamVid dataset. Results are reported in mean(std) format. The number of testing samples are listed in parentheses underneath the dataset name. UC denotes the uncalibrated result. ↓ denotes that lower is better. Best results are bolded and green indicates statistically significant differences w.r.t. FL+LTS. Note that due to GPU memory limits, results of MMCE and MMCE+LTS are for downsampled images, thus can not be directly compared with other methods. The goal of including them is to show that LTS can improve MMCE. LTS generally achieves the best performance on almost all metrics in the *All* region, *Boundary* region and *Local* region.

weighting schemes. For example, majority voting (MV) and plurality voting (PV) [21, 24] are the simplest ones that assume each atlas contributes with equal reliability to the estimate of the target segmentation, i.e. w_x^i is a constant value for all i and x. Moving forward, spatially varying weighted voting (SVWV) [2, 10, 61] relaxes the assumption to allow for spatially varying weights, i.e. w_x^i can be different for i and x. One simple way to estimate the weight w_x^i is to set it as the probability of $\tilde{A}_S^i(x) = T_S(x)$, i.e. $w_x^i = p(\tilde{A}_S^i(x) = T_S(x))$. Though SVWV significantly improves the performance over MV and PV, it fails to consider the situation that atlases may make correlated errors. Thus, joint label fusion (JLF) [64] has been proposed which down-weights pairs of atlases that consistently make similar errors. Specifically, JLF tries to find the optimal weights ω_x^i by minimizing the expected error between $\hat{T}_S(x)$ and the true segmentation $T_S(x)$:

$$E\left[(T_S(x) - \hat{T}_S(x))^2\right]. \tag{K.3}$$

Thus, label fusion weights can be computed from Eq. (K.4) by minimizing the total expectation of segmentation errors of Eq. (K.3) constrained to $\sum_{i=1}^n \omega_x^i = 1$:

$$\mathbf{w}_x = \frac{\mathbf{M}_x^{-1}\mathbf{1}_n}{\mathbf{1}_n^t \mathbf{M}_x^{-1}\mathbf{1}_n}, \tag{K.4}$$

where $\mathbf{1}_n$ is a vector of all 1 and t is the transpose. $\mathbf{w}_x$ is the vector of weights and w_x^i is its i-th entry (correspond to the i-th atlas). $\mathbf{M}_x$ is a pairwise dependency matrix of size $n \times n$ where each entry $\mathbf{M}_x(i,j)$ is the estimated joint probability that atlas $\tilde{A}_S^i$ (row) and $\tilde{A}_S^j$ (column) both provide wrong label suggestions for the target image T_I at location x. $\mathbf{M}_x(i,j)$ is approximated as follows:

$$\begin{aligned}
\mathbf{M}_x(i,j) &= p(\tilde{A}_S^i(x) \neq T_S(x), \tilde{A}_S^j(x) \neq T_S(x)) \\
&\approx p(\tilde{A}_S^i(x) \neq T_S(x)) p(\tilde{A}_S^j(x) \neq T_S(x)) \\
&= (1 - p(\tilde{A}_S^i(x) = T_S(x)))(1 - p(\tilde{A}_S^j(x) = T_S(x))).
\end{aligned} \tag{K.5}$$

Based on the above-mentioned label fusion approaches, the segmentation accuracy of MAS relies heavily on the accuracy of estimating the probability of the i-th atlas having the same label as the target image, i.e. $p(\tilde{A}_S^i(x) = T_S(x))$. Estimation of

$p(\tilde{A}_S^i(x) = T_S(x))$ is rarely explored. Typically, patch-based sum of squared differences (SSD) between image intensities are used [2, 10, 61, 64]. Recently, deep convolutional networks based approaches [12, 13, 66] have been proposed to improve over the SSD intensity measures and have achieved great success. Here, specifically, we employ a deep convolutional neural network called VoteNet+ [13] to estimate the probabilities. We then conduct experiments for probability calibration to determine how much improving the calibration can improve the joint label fusion result and in turn the segmentation accuracy.