Journal of Machine Learning Research 24 (2023) 1-50 Submitted 6/22; Revised 5/23; Published 5/23

RankSEG: A Consistent Ranking-based Framework for Segmentation

Ben Dai BENDAI@CUHK.EDU.HK
Department of Statistics
The Chinese University of Hong Kong
Hong Kong SAR

Chunlin Li LI000007@UMN.EDU
School of Statistics
University of Minnesota
MN 55455 USA

Editor: Christoph Lampert

Abstract

Segmentation has emerged as a fundamental field of computer vision and natural language processing, which assigns a label to every pixel/feature to extract regions of interest from an image/text. To evaluate the performance of segmentation, the Dice and IoU metrics are used to measure the degree of overlap between the ground truth and the predicted segmentation. In this paper, we establish a theoretical foundation of segmentation with respect to the Dice/IoU metrics, including the Bayes rule and Dice-/IoU-calibration, analogous to classification-calibration or Fisher consistency in classification. We prove that the existing thresholding-based framework with most operating losses are not consistent with respect to the Dice/IoU metrics, and thus may lead to a suboptimal solution. To address this pitfall, we propose a novel consistent ranking-based framework, namely *RankDice*/*RankIoU*, inspired by plug-in rules of the Bayes segmentation rule. Three numerical algorithms with GPU parallel execution are developed to implement the proposed framework in large-scale and high-dimensional segmentation. We study statistical properties of the proposed framework. We show it is Dice-/IoU-calibrated, and its excess risk bounds and the rate of convergence are also provided. The numerical effectiveness of *RankDice*/*mRankDice* is demonstrated in various simulated examples and *Fine-annotated CityScapes*, *Pascal VOC* and *Kvasir-SEG* datasets with state-of-the-art deep learning architectures. Python module and source code are available on GITHUB at `https://github.com/statmlben/rankseg`.

Keywords: Segmentation, Bayes rule, ranking, Dice-calibrated, excess risk bounds, Poisson-binomial distribution, normal approximation, GPU computing

1. Introduction

Segmentation is one of the key tasks in the field of computer vision and natural language processing, which groups together similar pixels/features of an input that belong to the same class (Ronneberger et al., 2015; Badrinarayanan et al., 2017). It has become an essential part of image and text understanding with applications in autonomous vehicles (Assidiq et al., 2008), medical image diagnostics (Wang et al., 2018), face/fingerprint recognition (Xin et al., 2018), and video action localization (Shou et al., 2017).

The primary aim of segmentation is to label each foreground feature/pixel of an input with a corresponding class. Specifically, for a feature vector or an image $\mathbf{X} \in \mathbb{R}^d$, a *segmentation function* $\boldsymbol{\delta}$:

©2023 Ben Dai and Chunlin Li.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v24/22-0712.html`.

$\mathbb{R}^d \to \{0,1\}^d$ yields a predicted segmentation $\boldsymbol{\delta}(\mathbf{X}) = (\delta_1(\mathbf{X}), \cdots, \delta_d(\mathbf{X}))^\intercal$, where $\delta_j(\mathbf{X})$ represents the predicted segmentation for the j-th feature X_j, and $I(\boldsymbol{\delta}(\mathbf{X})) = \{j : \delta_j(\mathbf{X}) = 1; \text{ for } j = 1, \cdots, d\}$ is the index set of the segmented features of $\mathbf{X}$ provided by $\boldsymbol{\delta}$. Correspondingly, $\mathbf{Y} \in \{0,1\}^d$ is a feature-wise label of a ground truth segmentation, where $Y_j = 1$ indicates that the j-th feature/pixel X_j is segmented, and $I(\mathbf{Y}) = \big\{j : \mathbf{Y}_j = 1; \text{ for } j = 1, \cdots, d\big\}$ is the index set of the ground-truth features.

To access the performance for a segmentation function $\boldsymbol{\delta}$, the Dice and IoU metrics are introduced and widely used in the literature (Milletari et al., 2016), both of which measure the overlap between the ground truth and the predicted segmentation:

$$\begin{aligned} \mathrm{Dice}_\gamma(\boldsymbol{\delta}) &= \mathbb{E}\Big(\frac{2\big|I(\mathbf{Y}) \cap I(\boldsymbol{\delta}(\mathbf{X}))\big| + \gamma}{|I(\mathbf{Y})| + |I(\boldsymbol{\delta}(\mathbf{X}))| + \gamma}\Big) = \mathbb{E}\Big(\frac{2\mathbf{Y}^\intercal \boldsymbol{\delta}(\mathbf{X}) + \gamma}{\|\mathbf{Y}\|_1 + \|\boldsymbol{\delta}(\mathbf{X})\|_1 + \gamma}\Big), \\ \mathrm{IoU}_\gamma(\boldsymbol{\delta}) &= \mathbb{E}\Big(\frac{\big|I(\mathbf{Y}) \cap I(\boldsymbol{\delta}(\mathbf{X}))\big| + \gamma}{\big|I(\mathbf{Y}) \cup I(\boldsymbol{\delta}(\mathbf{X}))\big| + \gamma}\Big) = \mathbb{E}\Big(\frac{\mathbf{Y}^\intercal \boldsymbol{\delta}(\mathbf{X}) + \gamma}{\|\mathbf{Y}\|_1 + \|\boldsymbol{\delta}(\mathbf{X})\|_1 - \mathbf{Y}^\intercal \boldsymbol{\delta}(\mathbf{X}) + \gamma}\Big), \end{aligned} \tag{1}$$

where $|\cdot|$ is the cardinality of a set, and $\gamma \geq 0$ is a smoothing parameter. When $\gamma = 0$, $\mathrm{Dice}_\gamma(\boldsymbol{\delta}) = \mathbb{E}\big(2\mathrm{TP}/(2\mathrm{TP} + \mathrm{FP} + \mathrm{FN})\big)$, $\mathrm{IoU}_\gamma(\boldsymbol{\delta}) = \mathbb{E}\big(\mathrm{TP}/(\mathrm{TP} + \mathrm{FP} + \mathrm{FN})\big)$ where TP is the true positive, FP is the false positive, and FN is the false negative. Both metrics are similar and will be treated in a unified fashion; however, as to be seen in the sequel, searching for the optimal segmentation function with respect to the IoU metric may require extra computation than its Dice counterpart. Thus, for clarity of presentation, we first focus on the Dice metric and postpone the discussion on the relationships between the Dice and IoU metrics in Section 4.2.

The recent success of fully convolutional networks has enabled significant progress in segmentation. In literature, the mainstream of recent works devoted to designing and developing neural network architectures under different segmentation scenarios, including FCN (Long et al., 2015), U-Net (Ronneberger et al., 2015), DeepLab (Chen et al., 2018), and PSPNet (Zhao et al., 2017). Despite their remarkable performance, most existing models primarily focus on predicting segmentation using a classification framework, without considering the inherent disparities between classification and segmentation (as discussed in Section 1.1). We find this framework leading to an inconsistent solution and suboptimal performance with respect to the Dice/IoU metrics, and we address this pitfall by developing a novel consistent ranking-based framework, namely *RankSEG* (*RankDice* to the Dice metric and *RankIoU* to the IoU metric), to improve the segmentation performance.

1.1 Existing methods

Most existing segmentation methods are developed under a threshold-based framework with two types of loss functions.

As indicated in Figure 1, the existing threshold-based segmentation framework, inspired by binary classification, provides a predicted segmentation via a two-stage procedure: (i) estimating a decision function or a probability function based on a loss; (ii) predicting feature-wise labels by thresholding the estimated decision function or probabilities. Specifically, given a training dataset $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$, the prediction provided by the threshold-based framework for a new instance $\mathbf{x}$ can be formulated as:

$$\widehat{\mathbf{q}} = \operatorname*{argmin}_{\mathbf{q} \in \mathcal{Q}} \frac{1}{n} \sum_{i=1}^n l\big(\mathbf{y}_i, \mathbf{q}(\mathbf{x}_i)\big) + \lambda \|\mathbf{q}\|^2, \qquad \widehat{\boldsymbol{\delta}}(\mathbf{x}) = \mathbf{1}(\widehat{\mathbf{q}}(\mathbf{x}) \geq 0.5), \tag{2}$$

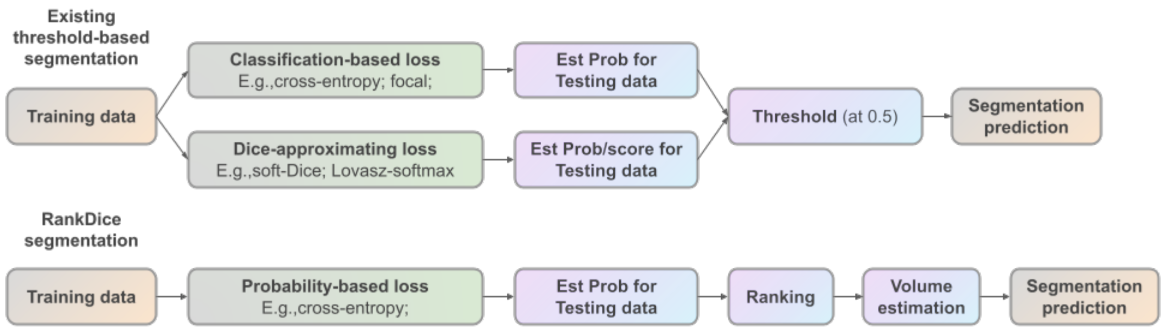
Existing threshold-based segmentation
Training data
Classification-based loss E.g.,cross-entropy; focal;
Dice-approximating loss E.g.,soft-Dice; Lovasz-softmax
Est Prob for Testing data
Est Prob/score for Testing data
Threshold (at 0.5)
Segmentation prediction
RankDice segmentation
Training data
Probability-based loss E.g.,cross-entropy;
Est Prob for Testing data
Ranking
Volume estimation
Segmentation prediction

Figure 1: The existing and the proposed (*RankDice*) frameworks for segmentation. The upper panel is the existing *threshold-based segmentation* framework, and the lower panel is the proposed *RankDice* framework.

where $l(\cdot,\cdot)$ is an operating loss, $\mathbf{q} : \mathbb{R}^d \to [0,1]^d$ is a candidate probability function with q_j being the candidate probability of the j-th pixel, $\mathcal{Q}$ is a class of candidate probability functions, $\|\mathbf{q}\|$ is a regularization term, $\lambda > 0$ is a tuning parameter to balance the overfitting and underfitting, and $\mathbf{1}(\cdot)$ is an indicator function. For ease of presentation, $q_j(\mathbf{x})$ is specified as a probability function and a predicted segmentation is produced by thresholding at 0.5, yet it can be equally extended to a general decision function. For example, we may formulate $q_j(\mathbf{x})$ as a decision function with range in $\mathbb{R}$, and the prediction is produced by thresholding at 0, analogous to SVMs in classification (Cortes and Vapnik, 1995). Next, under the framework (2), two different types of operating loss functions are considered, namely the *classification-based losses* and the *Dice-approximating losses*.
Classification-based losses completely characterize segmentation as a classification problem, with examples such as the cross-entropy (CE; Cox (1958)) and the focal loss (Focal; Lin et al. (2017)):

$$\text{(CE)} \qquad l_{\text{CE}}\big(\mathbf{y},\mathbf{q}(\mathbf{x})\big) = -\sum_{j=1}^{d}\Big(y_j \log\big(\mathbf{q}_j(\mathbf{x})\big) + (1-y_j)\log\big(1-\mathbf{q}_j(\mathbf{x})\big)\Big), \tag{3}$$

$$\text{(Focal)} \quad l_{\text{focal}}\big(\mathbf{y},\mathbf{q}(\mathbf{x})\big) = -\sum_{j=1}^{d}\Big(y_j(1-\mathbf{q}_j(\mathbf{x}))^{\vartheta}\log\big(\mathbf{q}_j(\mathbf{x})\big) + (1-y_j)\mathbf{q}_j^{\vartheta}(\mathbf{x})\log\big(1-\mathbf{q}_j(\mathbf{x})\big)\Big),$$

where $\vartheta \geq 0$ is a hyperparameter for the focal loss (Lin et al., 2017). Other margin-based losses such as the hinge loss, in principle, can be included as classification-based losses with a decision function ranged in $\mathbb{R}$ thresholding at 0, although they are less frequently used in a multiclass problem (Tewari and Bartlett, 2007). Therefore, we focus on the probability-based classification loss in the sequel.
Dice-approximating losses aim to approximate the Dice/IoU metric and conduct a direct optimization. Typical examples are the soft-Dice (Sudre et al., 2017) and the Lovasz-softmax loss (Berman et al., 2018):

$$\text{(Soft-Dice)} \qquad l_{\text{SoftD}}\big(\mathbf{y},\mathbf{q}(\mathbf{x})\big) = 1 - \frac{2\mathbf{y}^{\intercal}\mathbf{q}(\mathbf{x})}{\|\mathbf{y}\|_1 + \|\mathbf{q}(\mathbf{x})\|_1},$$

$$\text{(Lovasz-softmax)} \qquad l_{\text{l-softmax}}\big(\mathbf{y},\mathbf{q}(\mathbf{x})\big) = \overline{V}\big(\mathbf{y}\circ(1-\mathbf{q}(\mathbf{x})) + (1-\mathbf{y})\circ\mathbf{q}(\mathbf{x})\big),$$

where $\overline{V}(\cdot)$ is the Lovasz extension of the mis-IoU error (Berman et al., 2018), and $\circ$ is the element-wise product. Specifically, the soft-Dice loss replaces the binary segmentation indicator $\boldsymbol{\delta}(\mathbf{x}) \in$

$\{0,1\}^d$ in the Dice metric by a candidate probability function $\mathbf{q}(\mathbf{x}) \in [0,1]^d$ to make the computation feasible. The Lovasz-softmax directly takes a convex extension of IoU based on a softmax transformation. Moreover, other losses including the Tversky loss (Salehi et al., 2017), the Lovasz-hinge loss (Berman et al., 2018), and the log-Cosh Dice loss (Jadon, 2021) can be also categorized as Dice-approximating losses.

The threshold-based framework (2) with a classification-based loss or a Dice-approximating loss is a commonly used approach for segmentation. Although encouraging performance is delivered, we show that the minimizer from (2) (based on the cross-entropy, the soft-Dice and the focal loss) is **inconsistent** (or suboptimal) to the Dice metric, see Lemma 9. For Lovasz convex losses, it is still unclear if they are able to yield an optimal segmentation (convex closure is usually not enough to ensure the consistency (Bartlett et al., 2006)). Moreover, in practice, only a small number of pixel predictions are taken into account in one stochastic gradient step. Therefore, the Lovasz-softmax loss cannot directly optimize the segmentation metric (see Section 3.1 in Berman et al. (2018)).

Another line of work (Bao and Sugiyama, 2020; Nordström et al., 2020; Lipton et al., 2014) has been centered on a linear-fractional approximation of the Dice and IoU metrics which are not decomposable per instance. In particular, Bao and Sugiyama (2020) and Nordström et al. (2020) proposed to approximate the utility by tractable surrogate functions with a sample-splitting procedure and showed that their methods are consistent in optimizing the target utility. Yet, splitting the sample may undermine the efficiency. Lipton et al. (2014) indicated that the optimal threshold maximizing the F_1 metric is equal to half of the optimal F_1 value. However, their definitions of Dice and IoU metrics may overlook small instances, which is undesirable in many applications; see Appendix A for technical discussion.

To summarize, the current frameworks with existing losses may either yield a suboptimal solution or suffer from an inappropriate target metric, demanding efforts to further improve the performance, robustness and sustainability of the existing segmentation framework.

1.2 Our contribution

In this paper, we propose a novel Dice-calibrated ranking-based segmentation framework, namely *RankDice*, to address the inconsistency of the existing framework. *RankDice* is primarily inspired by the Bayes rule of Dice-segmentation. We summarize our major contribution as follows:

1. To our best knowledge, the proposed ranking-based segmentation framework *RankDice*, is the first consistent segmentation framework with respect to the Dice metric (Dice-calibrated).

2. Three numerical algorithms with GPU parallel execution are developed to implement the proposed framework in large-scale and high-dimensional segmentation.

3. We establish a theoretical foundation of segmentation with respect to the Dice metric, such as the Bayes rule and Dice-calibration. Moreover, we present Dice-calibrated consistency (Lemma 10) and a convergence rate of the excess risk (Theorem 11) for the proposed *RankDice* framework, and indicate inconsistent results for the existing methods (Lemma 9).

4. Our experiments in three simulated examples and three real datasets (CityScapes dataset, Pascal VOC 2021 dataset, and Kvasir-SEG dataset) suggest that the improvement of *RankDice* over the existing framework is significant for various loss functions and network architectures (see Tables 9-6).

It is worth noting that the results are equally applicable to the proposed *RankIoU* framework in terms of the IoU metric.

2. RankDice

It is reasonable to assume that all information on a feature-wise label is solely based on input features, that is, $Y_i \perp Y_j | \mathbf{X}$ for any $i \neq j$. In Appendix B.3, we provide a probabilistic perspective to suggest the necessity of this assumption in segmentation tasks. Without loss of generality, we further assume that $p_j(\mathbf{X}) := \mathbb{P}(Y_j = 1 | \mathbf{X})$ are distinct for $j = 1, \cdots, d$ with probability one.

2.1 Bayes segmentation rule

To begin with, we call segmentation with respect to the Dice metric as "Dice-segmentation". Then, we discuss Dice-segmentation at the population level, and present its Bayes (optimal) segmentation rule in Theorem 1 akin to the Bayes classifier for classification.

Theorem 1 (The Bayes rule for Dice-segmentation) *A segmentation rule $\boldsymbol{\delta}^*$ is a global maximizer of* $\mathrm{Dice}_\gamma(\boldsymbol{\delta})$ *if and only if it satisfies that*

$$\delta_j^*(\mathbf{x}) = \begin{cases} 1 & \textit{if } p_j(\mathbf{x}) \textit{ ranks top } \tau^*(\mathbf{x}), \\ 0 & \textit{otherwise}. \end{cases}$$

The optimal volume (the optimum total number of segmented features) $\tau^*(\mathbf{x})$ is given as

$$\tau^*(\mathbf{x}) = \operatorname*{argmax}_{\tau \in \{0,1,\cdots,d\}} \Big(\sum_{j \in J_\tau(\mathbf{x})} \sum_{l=0}^{d-1} \frac{2}{\tau + l + \gamma + 1} p_j(\mathbf{x}) \mathbb{P}\big(\Gamma_{\text{-}j}(\mathbf{x}) = l\big) + \sum_{l=0}^{d} \frac{\gamma}{\tau + l + \gamma} \mathbb{P}\big(\Gamma(\mathbf{x}) = l\big) \Big), \quad (4)$$

where $J_\tau(\mathbf{x}) = \big\{ j : \sum_{j'=1}^{d} \mathbf{1}\big(p_{j'}(\mathbf{x}) \geq p_j(\mathbf{x})\big) \leq \tau \big\}$ is the index set of the τ-largest conditional probabilities with $J_0(\mathbf{x}) = \emptyset$, $\Gamma(\mathbf{x}) = \sum_{j=1}^{d} B_j(\mathbf{x})$, and $\Gamma_{\text{-}j}(\mathbf{x}) = \sum_{j' \neq j} B_{j'}(\mathbf{x})$ are Poisson-binomial random variables, and $B_j(\mathbf{x})$ is a Bernoulli random variable with the success probability $p_j(\mathbf{x})$. See the definition of the Poisson-binomial distribution in Appendix B.2.

Two remarkable observations emerge from Theorem 1. First, the Bayes segmentation operator can be obtained via a two-stage procedure: (i) ranking the conditional probability $p_j(\mathbf{x})$, and (ii) searching for the optimal volume of the segmented features $\tau(\mathbf{x})$. Second, both the Bayes segmentation rule $\boldsymbol{\delta}^*(\mathbf{x})$ and the optimal volume function $\tau^*(\mathbf{x})$ are achievable when the conditional probability $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_d(\mathbf{x}))^\intercal$ is well-estimated. Therefore, our proposed framework *RankDice* is directly inspired by a general *plug-in rule* of the Bayes segmentation rule.

Moreover, Lemma 9 (and examples provided in its proof) indicates that the segmentation rule produced by existing frameworks, such as the threshold-based framework, can significantly differ from the optimum in Theorem 1. In fact, Lemma 9 further proves that the soft-Dice, the cross-entropy loss, the focal loss, and even a general classification-calibrated loss (Zhang, 2004; Bartlett et al., 2006), are not Dice-calibrated. See the definition of Dice-calibrated (Definition 8), and the negative results for existing frameworks (Lemma 9) in Section 4.1. Besides, the Bayes rule for IoU-segmentation is presented in Lemma 13.

Remark 2 (Suboptimal of a fixed-thresholding framework) *The Bayes rule of Dice-segmentation can be also regarded as adaptive thresholding of conditional probabilities. Specifically, for each input* $\mathbf{x}$*, the optimal segmentation rule can be rewritten as:*

$$\delta_j^*(\mathbf{x}) = \mathbf{1}\big(p_j(\mathbf{x}) \geq p_{j_{\tau^*(\mathbf{x})}}(\mathbf{x})\big), \textit{ where } p_{j_{\tau^*(\mathbf{x})}}(\mathbf{x}) \textit{ is the top-}\tau^*(\mathbf{x}) \textit{ largest probability over } \mathbf{p}(\mathbf{x}).$$

Alternatively, Theorem 1 indicates that the Bayes rule for Dice-segmentation is unlikely to be obtained by a fixed thresholding framework, since the optimal threshold $p_{j_{\tau^*(\mathbf{x})}}(\mathbf{x})$ *varies greatly across different inputs. Therefore, (tuning a threshold on) a fixed thresholding-based framework leads to a suboptimal solution.*

Remark 2 also explains the heterogeneity of optimal thresholds in various datasets indicated in Bice et al. (2021), and the suboptimality of a fixed-thresholding framework is also empirically supported by Table 10 and Figure 7 for simulated examples, and Table 7 and Figure 4 for real datasets. Furthermore, the fact that fixed-thresholding is suboptimal for Dice-segmentation should be compared with the existing results in classification. For binary classification, the optimal threshold maximizing the F_1 metric is equal to half of the optimal F_1 value, which is fixed (Lipton et al., 2014). This disparity stems from a different definition of the Dice metric (or F_1) for binary classification, where F_1-score (for binary classification) can be regard as a linear fractional utility of Dice defined in (1); see more discussion in Appendix A.

2.2 Proposed framework

Suppose a training dataset $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$ is given, where $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{y}_i \in \{0,1\}^d$ are the input features and the true label for the i-th instance. Inspired by Theorem 1, we develop a ranking-based framework *RankDice* for Dice-segmentation (Steps 1-3).
Step 1 (Conditional probability estimation): Estimate the conditional probability based on logistic regression (the cross-entropy loss):

$$\widehat{\mathbf{q}}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{q} \in \mathcal{Q}} - \sum_{i=1}^n \sum_{j=1}^d \Big(y_{ij} \log\big(q_j(\mathbf{x}_i)\big) + (1 - y_{ij}) \log\big(1 - q_j(\mathbf{x}_i)\big) \Big) + \lambda \|\mathbf{q}\|^2, \tag{5}$$

where $\mathcal{Q}$ is a class of candidate probability functions, $\|\mathbf{q}\|$ is a regularization for a candidate function, and $\lambda > 0$ is a hyperparameter to balance the loss and regularization. For example, $\mathbf{q} \in \mathcal{Q}$ is usually a deep convolutional neural network for image segmentation, and $\|\mathbf{q}\|$ can be a matrix norm of weight matrices in the network.
Step 2 (Ranking): Given a new instance $\mathbf{x}$, rank its estimated conditional probabilities decreasingly, and denote the corresponding indices as $j_1, \cdots, j_d$, that is, $\widehat{q}_{j_1}(\mathbf{x}) \geq \widehat{q}_{j_2}(\mathbf{x}) \geq \cdots \geq \widehat{q}_{j_d}(\mathbf{x})$.
Step 3 (Volume estimation): From (4), we estimate the volume $\widehat{\tau}(\mathbf{x})$ by replacing the true conditional probability $\mathbf{p}(\mathbf{x})$ by the estimated one $\widehat{\mathbf{q}}(\mathbf{x})$:

$$\widehat{\tau}(\mathbf{x}) = \operatorname*{argmax}_{\tau \in \{0,\cdots,d\}} \sum_{s=1}^{\tau} \sum_{l=0}^{d-1} \frac{2}{\tau + l + \gamma + 1} \widehat{q}_{j_s}(\mathbf{x}) \mathbb{P}\big(\widehat{\Gamma}_{-j_s}(\mathbf{x}) = l\big) + \sum_{l=0}^{d} \frac{\gamma}{\tau + l + \gamma} \mathbb{P}\big(\widehat{\Gamma}(\mathbf{x}) = l\big), \tag{6}$$

where $\widehat{\Gamma}(\mathbf{x}) = \sum_{j=1}^d \widehat{B}_j(\mathbf{x})$ and $\widehat{\Gamma}_{-j_s}(\mathbf{x}) = \sum_{j \neq j_s} \widehat{B}_j(\mathbf{x})$ are Poisson-binomial random variables, and $\widehat{B}_j(\mathbf{x})$ are independent Bernoulli random variables with success probabilities $\widehat{q}_j(\mathbf{x})$; for $j = 1, \cdots, d$.

Finally, the predicted segmentation $\widehat{\boldsymbol{\delta}}(\mathbf{x}) = (\widehat{\delta}_1(\mathbf{x}), \cdots, \widehat{\delta}_d(\mathbf{x}))^\intercal$ is given by selecting the indices of the top-$\widehat{\tau}(\mathbf{x})$ conditional probabilities:

$$\widehat{\delta}_j(\mathbf{x}) = 1, \text{ if } j \in \{j_1, \cdots, j_{\widehat{\tau}(\mathbf{x})}\}; \quad \widehat{\delta}_j(\mathbf{x}) = 0, \text{ otherwise.} \tag{7}$$

The proposed *RankDice* framework (Steps 1-3) provides a feasible solution to the Bayes segmentation rule in terms of the Dice metric. Note that **Step 1** is a standard conditional probability estimation, and **Step 2** simply ranks the estimated conditional probabilities. Next, we focus on developing a scalable computing scheme for **Step 3**.

2.3 Scalable computing schemes

This section develops scalable computing schemes for *volume estimation* in (6). Note that (6) can be rewritten as:

$$\widehat{\tau}(\mathbf{x}) = \underset{\tau \in \{0,\cdots,d\}}{\operatorname{argmax}}\ \bar{\pi}_\tau(\mathbf{x}); \qquad \bar{\pi}_\tau(\mathbf{x}) = \bar{\omega}_\tau(\mathbf{x}) + \bar{\nu}_\tau(\mathbf{x}),$$
$$\bar{\omega}_\tau(\mathbf{x}) = \sum_{l=0}^{d-1} \frac{2\omega_{\tau,l}(\mathbf{x})}{\tau + l + \gamma + 1}, \quad \omega_{\tau,l}(\mathbf{x}) = \sum_{s=1}^{\tau} \widehat{q}_{j_s}(\mathbf{x}) \mathbb{P}(\widehat{\Gamma}_{-j_s}(\mathbf{x}) = l), \quad \bar{\nu}_\tau(\mathbf{x}) = \sum_{l=0}^{d} \frac{\gamma \mathbb{P}(\widehat{\Gamma}(\mathbf{x}) = l)}{\tau + l + \gamma}. \tag{8}$$

The computational complexity of solving (8) is intimately related to the dimension of input features. Therefore, we develop numerical algorithms for low- and high-dimensional segmentation separately.

2.3.1 EXACT ALGORITHMS FOR LOW-DIMENSIONAL SEGMENTATION

Exact algorithm based on FFT. When the dimension is low ($d \leq 500$), we consider an exact algorithm to evaluate $\bar{\omega}_\tau$ and $\bar{\nu}_\tau$. According to the definition of $\omega_{\tau,l}$, it can be computed by the following recursive formula ($\tau = 1, \cdots, d$):

$$\boldsymbol{\omega}_\tau(\mathbf{x}) = \boldsymbol{\omega}_{\tau-1}(\mathbf{x}) + \widehat{q}_{j_\tau}(\mathbf{x}) \big(\mathbb{P}(\widehat{\Gamma}_{-j_\tau}(\mathbf{x}) = 0), \cdots, \mathbb{P}(\widehat{\Gamma}_{-j_\tau}(\mathbf{x}) = d-1)\big)^\intercal, \quad \boldsymbol{\omega}_0(\mathbf{x}) = \mathbf{0}, \tag{9}$$

where $\boldsymbol{\omega}_\tau(\mathbf{x}) = (\omega_{\tau,0}(\mathbf{x}), \cdots, \omega_{\tau,d-1}(\mathbf{x}))^\intercal$. On this ground, it suffices to evaluate $\mathbb{P}(\widehat{\Gamma}(\mathbf{x}) = l)$ and $\mathbb{P}(\widehat{\Gamma}_{-j_\tau}(\mathbf{x}) = l)$, which are the probability mass functions of Poisson-binomial random variables $\widehat{\Gamma}(\mathbf{x})$ and $\widehat{\Gamma}_{-j_\tau}(\mathbf{x})$, respectively. As indicated in Hong (2013), they can be efficiently evaluated by a fast Fourier transformation (FFT). Based on the numerical results in Hong (2013), the computing time for FFT evaluation with $d \leq 500$ is generally negligible (less than ten milliseconds). Moreover, it is worth noting that our algorithm in (9) needs not store the entire auxiliary matrix $(\boldsymbol{\omega}_1(\mathbf{x}), \cdots, \boldsymbol{\omega}_d(\mathbf{x}))$, since the τ-th row $\boldsymbol{\omega}_\tau$ can be computed from the previous row $\boldsymbol{\omega}_{\tau-1}$. Hence, only $O(d)$ storage is required in (9). The detailed algorithm is summarized in Algorithm 1 with `approx=False`.

2.3.2 APPROXIMATION ALGORITHMS FOR HIGH-DIMENSIONAL SEGMENTATION

For high-dimensional segmentation, especially for image segmentation, it is challenging to solve (6) by a grid searching over $\tau \in \{0, \cdots, d\}$. To address this issue, we use shrinkage and approximation techniques to reduce the computational complexity of (6). First, Lemma 3 is developed to shrink the searching range of τ.

Lemma 3 (Shrinkage) *If* $\sum_{s=1}^{\tau}\widehat{q}_{j_s}(\mathbf{x}) \geq (\tau+\gamma+d)\widehat{q}_{j_{\tau+1}}(\mathbf{x})$*, then* $\bar{\pi}_\tau(\mathbf{x}) \geq \bar{\pi}_{\tau'}(\mathbf{x})$ *for all* $\tau' > \tau$*.*

Lemma 3 can be viewed as an early stopping of the grid searching, which draws from the property of Poisson binomial distribution (c.f. Lemma 17). Accordingly, we can shrink the grid search in (6) from $\{0,\cdots,d\}$ to $\{0,\cdots,d_0(\mathbf{x})\}$, which significantly reduces the computational complexity. Specifically,

$$\widehat{\tau}(\mathbf{x}) = \operatorname*{argmax}_{\tau\in\{0,\cdots,d_0(\mathbf{x})\}} \bar{\pi}_\tau(\mathbf{x}); \quad d_0(\mathbf{x}) = \min\big\{\tau = 1,\cdots,d : \sum_{s=1}^{\tau}\widehat{q}_{j_s}(\mathbf{x})/\widehat{q}_{j_{\tau+1}}(\mathbf{x}) \geq \tau+\gamma+d\big\}. \tag{10}$$

In many applications, $d_0(\mathbf{x})$ is upper bounded by a small integer, that is, $d_0(\mathbf{x}) < d_U \ll d$, called the *well-separated segmentation*. For example, there exists a small number of features/pixels whose probabilities (close to 1) are much larger than the others.

Truncated refined normal approximation (T-RNA). Note that the cumulative distribution function (CDF), and thus the probability mass function, of a Poisson-binomial random variable can be efficiently evaluated by a refined normal approximation when the dimension is large (Hong, 2013; Neammanee, 2005). For instance,

$$\begin{aligned}\mathbb{P}(\widehat{\Gamma}(\mathbf{x}) \leq l) \ &\leftarrow \ \widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) \leq l) := \Psi\big(\widehat{\sigma}^{-1}(l+1/2-\widehat{\mu}(\mathbf{x}))\big),\\ \mathbb{P}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) \leq l) \ &\leftarrow \ \widetilde{\mathbb{P}}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) \leq l) := \Psi_{\text{-}j}\big(\widehat{\sigma}_{\text{-}j}^{-1}(l+1/2-\widehat{\mu}_{\text{-}j}(\mathbf{x}))\big),\end{aligned} \tag{11}$$

where $\Psi(u) = \Phi(u) + \widehat{\eta}(\mathbf{x})(1-u^2)\phi(u)/6$ and $\Psi_{\text{-}j}(u) = \Phi(u) + \widehat{\eta}_{\text{-}j}(\mathbf{x})(1-u^2)\phi(u)/6$ are two skew-corrected refined normal CDFs, $\Phi(\cdot)$ is the CDF of the standard normal distribution, and $(\widehat{\mu}(\mathbf{x}),\widehat{\mu}_{\text{-}j}(\mathbf{x}))$, $(\widehat{\sigma}^2(\mathbf{x}),\widehat{\sigma}^2_{\text{-}j}(\mathbf{x}))$, $(\widehat{\eta}(\mathbf{x}),\widehat{\eta}_{\text{-}j}(\mathbf{x}))$ are mean, variance and skewness of the Poisson-binomial random variables $\widehat{\Gamma}(\mathbf{x})$ and $\widehat{\Gamma}_{\text{-}j}(\mathbf{x})$, respectively. See the definitions of variance and skewness of the Poisson-binomial distribution in Appendix B.2.

On this ground, it is unnecessary to compute all $\mathbb{P}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) = l)$ and $\mathbb{P}(\widehat{\Gamma}(\mathbf{x}) = l)$ for $l = 1,\cdots,d$, since they are negligibly close to zero when l is too small or too large. In other words, many $\mathbb{P}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) = l)$ and $\mathbb{P}(\widehat{\Gamma}(\mathbf{x}) = l)$ can be ignored when evaluating $\bar{\omega}_\tau$ and $\bar{\nu}_\tau$. Therefore, according to the refined normal approximation in (11), $\bar{\omega}_\tau$ and $\bar{\nu}_\tau$ can be approximated by only taking a partial sum over a subset of $l = 0,\cdots,d$:

$$\begin{aligned}&\widetilde{\omega}_\tau(\mathbf{x}) = \sum_{l\in\mathcal{L}(\varepsilon)} \frac{2\widetilde{\omega}_{\tau,l}(\mathbf{x})}{\tau+l+\gamma+1}, \quad \widetilde{\nu}_\tau(\mathbf{x}) = \sum_{l\in\mathcal{L}(\varepsilon)} \frac{\gamma\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l)}{\tau+l+\gamma}, \quad \widetilde{\omega}_{\tau,l}(\mathbf{x}) = \sum_{s=1}^{\tau}\widehat{q}_{j_s}(\mathbf{x})\widetilde{\mathbb{P}}(\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x}) = l),\\ &\mathcal{L}(\varepsilon) = \Big\{\lfloor\widehat{\sigma}(\mathbf{x})\Psi^{-1}(\varepsilon) + \widehat{\mu}(\mathbf{x}) - \frac{3}{2}\rfloor,\cdots,\lfloor\widehat{\sigma}(\mathbf{x})\Psi^{-1}(1-\varepsilon) + \widehat{\mu}(\mathbf{x}) - \frac{1}{2}\rfloor\Big\}\bigcap\{0,\cdots,d\},\end{aligned} \tag{12}$$

where $\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l) := \widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) \leq l) - \widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) \leq l-1)$, $\widetilde{\mathbb{P}}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) = l) := \widetilde{\mathbb{P}}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) \leq l) - \widetilde{\mathbb{P}}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) \leq l-1)$, $\Psi^{-1}(\cdot)$ is the quantile function of the refined normal distribution, and ε is a custom tolerance error. The detailed algorithm is summarized in Algorithm 1 with `approx='T-RNA'`. Moreover, the following lemma quantifies the approximation error of the proposed approximate algorithm.

Lemma 4 *For* $(\bar{\omega}_\tau,\bar{\nu}_\tau)$ *and* $(\widetilde{\omega}_\tau,\widetilde{\nu}_\tau)$ *defined in* (8) *and* (12) *respectively, if* $\widehat{\sigma}^2(\mathbf{x}) \geq 25$*, then for any* $\tau \in \{0,\cdots,d\}$*, we have*

$$\begin{aligned}|\widetilde{\omega}_\tau - \bar{\omega}_\tau| &\leq \frac{4\tau}{\tau+\gamma+1}\big(\varepsilon + \frac{C_0}{\widehat{\sigma}^2(\mathbf{x})}\big) + \frac{C_0\min(\widehat{\mu}(\mathbf{x}),\tau)}{\widehat{\sigma}^2(\mathbf{x})-1/4}\Big(\log\big(1+d\big)+1\Big),\\ |\widetilde{\nu}_\tau - \bar{\nu}_\tau| &\leq \frac{2\gamma}{\tau+\gamma}\big(\varepsilon + \frac{C_0}{\widehat{\sigma}^2(\mathbf{x})}\big) + \frac{C_0\gamma}{\widehat{\sigma}^2(\mathbf{x})}\Big(\log\big(1+d\big)+1\Big),\end{aligned}$$

where $C_0 = 0.1618$ *if* $\widehat{\sigma}^2(\mathbf{x}) \geq 100$*, and* $C_0 = 0.3056$ *if* $\widehat{\sigma}^2(\mathbf{x}) \geq 25$*. Moreover, when* $d \to \infty$*, we have*

$$|\widetilde{\pi}_\tau - \overline{\pi}_\tau| \leq \big(\frac{4\tau}{\tau+\gamma+1} + \frac{2\gamma}{\tau+\gamma}\big)\varepsilon + O\big(\frac{\min(\widehat{\mu}(\mathbf{x}), \tau)\log(d)}{\widehat{\sigma}^2(\mathbf{x})}\big).$$

Here we define $0/0 := 0$ *for* $\gamma/(\tau+\gamma)$ *when* $\tau = \gamma = 0$*.*

Note that $\widehat{\sigma}^2(\mathbf{x}) = \sum_{j=1}^d \widehat{q}_j(\mathbf{x})(1-\widehat{q}_j(\mathbf{x}))$ is the variance of $\widehat{\Gamma}(\mathbf{x})$, which often tends to infinity as $d \to \infty$. In image segmentation, the dimension d varies from 1024 (32x32) to 262144 (512x512). Therefore, the error bound is practical for high-dimensional segmentation. Moreover, ε is the tolerance error to balance the approximation error and computation complexity. For instance, when ε becomes smaller, the approximation error decreases, the computation complexity increases with an enlarged $\mathcal{L}(\varepsilon)$. Typically, $\Psi^{-1}(1-\varepsilon) - \Psi^{-1}(\varepsilon) \leq 7.438$ when $\varepsilon = 1\text{e}{-4}$. Based on the approximation formula (12), the worst-case computational complexity is reduced to $O(d\widehat{\sigma}(\mathbf{x}))$ for general segmentation and $O(\widehat{\sigma}(\mathbf{x}))$ for *well-separated segmentation* ($d_0(\mathbf{x}) \leq d_U$). The computational complexity is summarized in Table 1 (based on $\varepsilon = 1\text{e}{-4}$).

Although T-RNA significantly reduces the computational complexity, yet this algorithm uses recursive updates, making it difficult in parallel computing on GPUs. For example, due to the memory issues, recursive function calls are restricted in the CUDA (Compute Unified Device Architecture) kernels (Nickolls et al., 2008). Next, we propose the blind approximation algorithm to exploit GPU-computing for the proposed *RankDice*.

Blind approximation (BA). In high-dimensional segmentation, the difference in distributions between $\widehat{\Gamma}(\mathbf{x})$ and $\widehat{\Gamma}_{-j}(\mathbf{x})$ is negligible. Inspired by this fact, we propose a novel approximation algorithm, namely the blind approximation, to simultaneously evaluate the score functions with a small error tolerance. Specifically, based on the refined normal approximation, we further simplify the evaluation formulas by replacing $\widetilde{\mathbb{P}}(\widehat{\Gamma}_{-j_s}(\mathbf{x}) = l)$ as $\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l)$:

$$\widetilde{\omega}^b_\tau(\mathbf{x}) = 2\sum_{s=1}^{\tau} \widehat{q}_{j_s}(\mathbf{x}) \sum_{l \in \mathcal{L}(\varepsilon)} \frac{\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l)}{\tau + l + \gamma + 1}, \quad \widetilde{\nu}^b_\tau(\mathbf{x}) = \widetilde{\nu}_\tau(\mathbf{x}) = \sum_{l \in \mathcal{L}(\varepsilon)} \frac{\gamma \widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l)}{\tau + l + \gamma},$$

where $\mathcal{L}(\varepsilon) = \{l_L, \cdots, l_U\}$ is defined in (12). Then, for any $1 \leq d_0 \leq d$, $\widetilde{\boldsymbol{\omega}}^b_{1:T} = (\widetilde{\omega}^b_1, \cdots, \widetilde{\omega}^b_T)^\intercal$ and $\widetilde{\boldsymbol{\nu}}^b_{1:T} = (\widetilde{\nu}^b_1, \cdots, \widetilde{\nu}^b_T)^\intercal$ can be simultaneously computed over $\tau = 0, \cdots, d_0$ based on cross-correlation (flipped convolution (Tahmasebi et al., 2012)):

$$\begin{aligned}\widetilde{\boldsymbol{\omega}}^b_{1:d_0} &= 2\widehat{\mathbf{s}}_{1:d_0}(\mathbf{x}) \circ \Big(\big(\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l_L), \cdots, \widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l_U)\big)^\intercal \star \big(\frac{1}{l_l + \gamma + 1}, \cdots, \frac{1}{l_U + \gamma + d_0 + 1}\big)\Big), \\ \widetilde{\boldsymbol{\nu}}^b_{1:d_0} &= \gamma\Big(\big(\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l_L), \cdots, \widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x}) = l_U)\big)^\intercal \star \big(\frac{1}{l_l + \gamma}, \cdots, \frac{1}{l_U + \gamma + d_0}\big)\Big),\end{aligned} \tag{13}$$

where $\widehat{\mathbf{s}}(\mathbf{x}) = (\widehat{s}_1(\mathbf{x}), \cdots, \widehat{s}_d(\mathbf{x}))^\intercal$ and $\widehat{s}_K(\mathbf{x}) = \sum_{k=1}^K \widehat{q}_{j_k}(\mathbf{x})$ is the cumulative sum of sorted estimated probabilities, $\circ$ is element-wise product of two vectors, and $\star$ is the cross-correlation operator (flipped convolution) of two vectors (Tahmasebi et al., 2012). Notably, the cross-correlation can be efficiently implemented by a fast Fourier transform (Bracewell and Bracewell, 1986) with time complexity $O((d_0 + \widehat{\sigma}(\mathbf{x}))\log(d_0 + \widehat{\sigma}(\mathbf{x})))$. Besides, the overall time complexity with CUDA implementation on GPUs can be further reduced by parallel computing. The detailed algorithm is summarized in Algorithm 1 with `approx='BA'`. Next, Lemma 5 shows the approximation error of the proposed blind approximation algorithm.

Algorithm 1: Computing schemes for the proposed RankDice framework.

```
Input  : Training set: $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$; A new testing instance: $\mathbf{x}$; Approx method: approx
Output: The predicted segmentation for the testing instance $\widehat{\boldsymbol{\delta}}(\mathbf{x})$
1  Conditional prob est. Estimate conditional prob $\widehat{\mathbf{q}}$ via (5) based on training set $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$;
2  Ranking. Rank estimated conditional probabilities for $\mathbf{x}$, and denote the indices in sorted
     order as $j_1, \cdots, j_d$, that is, $\widehat{q}_{j_1}(\mathbf{x}) \geq \widehat{q}_{j_2}(\mathbf{x}) \geq \cdots \geq \widehat{q}_{j_d}(\mathbf{x})$;
3  Volume estimation.
4  Compute and store the cumulative sum of sorted estimated probabilities $\widehat{\mathbf{s}}(\mathbf{x})$;
5  Compute $d_0(\mathbf{x})$ based on $\widehat{\mathbf{s}}(\mathbf{x})$ via (10);
6  if approx is None then
7      $\mathcal{L} = \{0, \cdots, d\}$;
8  else
9      $\mathcal{L} = \mathcal{L}(\varepsilon) = \{l_L, \cdots, l_U\}$ based on (12);
10 end
11 Compute and store $\mathbb{P}(\widehat{\Gamma}(\mathbf{x}) = l)$ for $l \in \mathcal{L}$;
12 if approx is BA then
13     Compute and store $\widetilde{\boldsymbol{\omega}}^b_{1:d_0(\mathbf{x})}$ and $\widetilde{\boldsymbol{\nu}}^b_{1:d_0(\mathbf{x})}$ based on (13);
14 else
15     Initialize $\boldsymbol{\omega}_{old} = \mathbf{0}$;
16     for $\tau = 1, \cdots, d_0(\mathbf{x})$ do
17         Update $\boldsymbol{\omega}_{new}$ as
               $\omega_{new,l} \leftarrow \omega_{old,l} + \widehat{q}_{j_\tau}(\mathbf{x})\mathbb{P}(\widehat{\Gamma}_{-j_\tau}(\mathbf{x}) = l), \text{ for } l \in \mathcal{L}, \quad \boldsymbol{\omega}_{old} \leftarrow \boldsymbol{\omega}_{new},$
           where $\widehat{\Gamma}_{-j}(\mathbf{x})$ is Poisson binomial r.v. with the success probabilities $\widehat{\mathbf{q}}_{-j}(\mathbf{x})$;
18         Compute and store $\bar{\pi}_\tau = \bar{\omega}_\tau + \bar{\nu}_\tau = \sum_{l \in \mathcal{L}} \frac{1}{\tau+l+1} \omega_{new,l} + \sum_{l \in \mathcal{L}} \frac{\gamma}{\tau+l+\gamma} \mathbb{P}(\widehat{\Gamma}(\mathbf{x}) = l)$;
19     end
20 end
21 Estimate $\widehat{\tau}(\mathbf{x}) = \text{argmax}_{\tau=0,\cdots,d_0(\mathbf{x})} \bar{\pi}_\tau$ via (8);
22 Prediction. The predicted segmentation is provided as:
       $\widehat{\delta}_j(\mathbf{x}) = 1, \text{ if } j \in \{j_1, \cdots, j_{\widehat{\tau}(\mathbf{x})}\}; \quad \widehat{\delta}_j(\mathbf{x}) = 0, \text{ otherwise.}$
   return The predicted segmentation $\widehat{\boldsymbol{\delta}}(\mathbf{x}) = (\widehat{\delta}_1(\mathbf{x}), \cdots, \widehat{\delta}_d(\mathbf{x}))^\intercal$.
```

Lemma 5 *For* $(\overline{\omega}_\tau, \overline{\nu}_\tau)$ *and* $(\widetilde{\omega}^b_\tau, \widetilde{\nu}^b_\tau)$ *defined in* (8) *and* (13) *respectively, if* $\widehat{\sigma}^2(\mathbf{x}) \geq 25$*, then for any* $\tau \in \{0, \cdots, d\}$*, and any* $\gamma \geq 0$*, we have*

$$\big|\widetilde{\omega}^b_\tau - \overline{\omega}_\tau\big| \leq \frac{4\tau}{\tau+\gamma+1}(\varepsilon + \frac{C_0}{\widehat{\sigma}^2(\mathbf{x})}) + \frac{C_0 \min(\widehat{\mu}(\mathbf{x}), \tau)}{\widehat{\sigma}^2(\mathbf{x}) - 1/4}\Big(\log\big(1+d\big)+1\Big)$$
$$+ \frac{1}{4\sqrt{2\pi}}\Big(\frac{1/(2\sqrt{e})}{\widehat{\sigma}^2(\mathbf{x}) - 1/4} + \frac{4}{\sqrt{\widehat{\sigma}^2(\mathbf{x}) - 1/4}} + \frac{4\widehat{m}_3(\mathbf{x})}{(\widehat{\sigma}^2(\mathbf{x}) - 1/4)^{3/2}}\Big)\Big(\log\big(1+d\big)+1\Big),$$

where $\widehat{m}_3(\mathbf{x}) = \sum_{j=1}^d \widehat{q}_j(\mathbf{x})(1-\widehat{q}_j(\mathbf{x}))(1-2\widehat{q}_j(\mathbf{x}))$*,* $C_0 = 0.1618$ *if* $\widehat{\sigma}^2(\mathbf{x}) \geq 100$*, and* $C_0 = 0.3056$ *if* $\widehat{\sigma}^2(\mathbf{x}) \geq 25$*. Specifically, when* $d \to \infty$*, we have*

$$|\widetilde{\pi}_\tau - \overline{\pi}_\tau| \leq (\frac{4\tau}{\tau+\gamma+1} + \frac{2\gamma}{\tau+\gamma})\varepsilon + O\Big((\frac{\min(\widehat{\mu}(\mathbf{x}), \tau)}{\widehat{\sigma}^2(\mathbf{x})} + \frac{1}{\widehat{\sigma}(\mathbf{x})})\log(d)\Big).$$

In contrast to the truncated refined normal approximation, the blind approximation algorithm significantly reduces the time complexity and enables GPU parallel execution. On the other hand, the blind approximation leads to an additional $\widehat{\sigma}^{-1}(\mathbf{x})$ in Lemma 5 compared with that of T-RNA in Lemma 4, yet the error bound is still acceptable in practice.

Taken together, we summarize the foregoing computational schemes in Algorithm 1, and their inference (after obtaining conditional probabilities) computational complexity (worst-case) is indicated in Table 1. For the threshold-based framework, thresholding the estimated probabilities takes $O(d)$ time complexity. For the proposed method, Step 2 takes $O(d\log(d))$ based on the merge sort (Ajtai et al., 1983), and Step 3 takes $O(d_0(\mathbf{x})\widehat{\sigma}(\mathbf{x}))$ based on T-RNA in (10) and (11), and takes $O\big(d_0(\mathbf{x})\log(d_0(\mathbf{x}))\big)$ based on BA in (13).

SEG framework	Time	Time (well-separated)	Memory
Threshold-based SEG	$O(d)$	$O(d)$	$O(d)$
RankDice (our)	$O(d\log(d) + dd_0(\mathbf{x}))$	$O(d\log(d))$	$O(d)$
RankDice-RNA (our)	$O(d\log(d) + \widehat{\sigma}(\mathbf{x})d_0(\mathbf{x}))$	$O(d\log(d))$	$O(d)$
RankDice-BA (our)	$O(d_0(\mathbf{x})\log(d_0(\mathbf{x})))$	$O(d\log(d))$	$O(d)$

Table 1: Inference (prediction) computational complexity for the segmentation frameworks. Here "Time" denotes the time complexity, "Memory" denotes the amount of storage needed including probability outcomes, "Time (well-separated)" denotes the time complexity on *well-separated segmentation* ($d_0(\mathbf{x}) \leq d_U$), and $d_0(\mathbf{x}) \leq d$ is the reduced dimension defined in (10), $\widehat{\sigma}(\mathbf{x})$ is the standard deviation of $\widehat{\Gamma}(\mathbf{x})$ with at most an order of $O(\sqrt{d})$. Note that the time complexity of RankDice-BA can be further reduced by GPU implementation. See more detailed discussion in Section 2.3.

3. mDice-segmentation and mRankDice

In this section, we discuss the extension of the proposed *RankDice* framework to segmentation with multiclass/multilabel outcomes evaluated by the mDice metric. Overall, multiclass/multilabel segmentation differs from Dice-segmentation in a number of important aspects. First, a new evaluation metric called mDice is introduced. Second, multiclass/multilabel outcomes provide two different ways of probabilistic modeling. Third, whether (or not) to allow for overlapping segmentation results among different classes leads to different problem setups. These aspects will be discussed in detail in the following subsections.

3.1 The mDice metric

For multiclass/multilabel segmentation, $(\mathbf{X}, \mathbf{Y}_{\cdot 1}, \cdots, \mathbf{Y}_{\cdot K})$ is available, where $\mathbf{X} \in \mathbb{R}^d$ represents a feature vector, K is the total number of classes of interest, $\mathbf{Y}_{\cdot k} \in \{0,1\}^d$ is the true feature-wise segmentation label for the k-th class, where $Y_{jk} = 1$ indicates that the j-th feature X_j is segmented under the k-th class, and $I(\mathbf{Y}_{\cdot k}) = \{j : Y_{jk} = 1;\text{ for } j = 1, \cdots, d\}$ is the class-specific index set for all segmented features.

On this ground, a class-specific segmentation operator $\mathbf{\Delta}_k : \mathbb{R}^d \to \{0,1\}^d (k = 1, \cdots, K)$ is introduced, where $\Delta_{jk}(\mathbf{x}) \in \{0,1\}$ is the predicted segmentation for the class k of the j-th feature, and $I(\mathbf{\Delta}_k(\mathbf{X})) = \{j : \Delta_{jk}(\mathbf{X}) = 1;\text{ for } j = 1, \cdots, d\}$ is the class-specific index set of features for the predicted segmentation. Then, given a segmentation operator $\mathbf{\Delta} = (\mathbf{\Delta}_1, \cdots, \mathbf{\Delta}_K)$, the multi-Dice (mDice) metric is defined as:

$$\begin{aligned}\mathrm{mDice}_\gamma(\mathbf{\Delta}) &= \sum_{k=1}^K \alpha_k \mathrm{Dice}_{\gamma,k}(\mathbf{\Delta}_k) \\ &= \sum_{k=1}^K \alpha_k \mathbb{E}\Big(\frac{2\big|I(\mathbf{Y}_{\cdot k}) \cap I(\mathbf{\Delta}_k(\mathbf{X}))\big| + \gamma}{|I(\mathbf{Y}_{\cdot k})| + |I(\mathbf{\Delta}_k(\mathbf{X}))| + \gamma}\Big) = \sum_{k=1}^K \alpha_k \mathbb{E}\Big(\frac{2\mathbf{Y}_{\cdot k}^\intercal \mathbf{\Delta}_k(\mathbf{X}) + \gamma}{\|\mathbf{Y}_{\cdot k}\|_1 + \|\mathbf{\Delta}_k(\mathbf{X})\|_1 + \gamma}\Big), \end{aligned} \tag{14}$$

where $\mathrm{Dice}_{\gamma,k}(\cdot)$ is the Dice metric under the k-th class, $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_K)^\intercal \geq \mathbf{0}_K$ is a weight vector for classes with $\|\boldsymbol{\alpha}\|_1 = 1$. For example, $\alpha_k = 1/K$ yields that each class has the same contribution to segmentation performance. More generally, $\boldsymbol{\alpha} \geq \mathbf{0}_K$ can be a custom weight vector indicating the relative importance of segmentation classes. In practice, given a new instance $(\mathbf{x}, \mathbf{y})$, the weight is an average over classes excluding those are not present and not predicted, that is,

$$\alpha_k = 0, \text{ if } \|\mathbf{y}_{\cdot k}\|_1 = \|\mathbf{\Delta}_k(\mathbf{x})\|_1 = 0; \qquad \alpha_k = \frac{1}{\sum_{k=1}^K \mathbf{1}(\|\mathbf{y}_{\cdot k}\|_1 + \|\mathbf{\Delta}_k(\mathbf{x})\|_1 > 0)}, \text{ otherwise.} \tag{15}$$

Following our convention, we shall call multiclass/multilabel segmentation with respect to the mDice metric as "mDice-segmentation". As indicated in Figure 2, unlike Dice-segmentation, mDice-segmentation provides more flexibility in probabilistic modeling (multiclass or multilabel) and the decision-making in prediction (taking argmax or thresholding at 0.5), resulting in different operating losses and the construction of predictive functions.

3.2 Multilabel/multiclass outcomes

In this section, we describe two probabilistic models (multilabel or multiclass) of $\mathbf{Y}_j | \mathbf{X}$, where $\mathbf{Y}_j = (Y_{j1}, \cdots, Y_{jK})^\intercal$ is the true label for the j-th feature.

For multilabel modeling, we assume each feature could be assigned to multiple classes, that is, $\|\mathbf{Y}_j\|_1 \in \{0, \cdots, K\}$, for $j = 1, \cdots, d$. As a result, the index sets of segmented features may overlap among classes. In this case, we formulate and estimate $q_{jk}(\mathbf{x})$ under a multilabel probabilistic model. For example, for deep learning models, we use the sigmoid function as the output layer activation function of a neural network with the *binary cross-entropy* loss.

For multiclass modeling, each feature is assigned to one and only one class, that is, $\|\mathbf{Y}_j\|_1 = 1$ for $j = 1, \cdots, d$. An instance is the panoptic annotation in image segmentation. In this case, we formulate and estimate $\mathbf{q}_j(\mathbf{x}) = (q_{j1}(\mathbf{x}), \cdots, q_{jK}(\mathbf{x}))^\intercal$ under a multiclass model with additional sum-to-one constraints $\|\mathbf{q}_j(\mathbf{x})\|_1 = 1$ for $j = 1, \cdots, d$ and $\mathbf{x} \in \mathbb{R}^d$. Specifically, for deep learning models, we use the softmax function as the output layer activation function of a neural network, which automatically enforces the sum-to-one constraints. Correspondingly, a multiclass loss is used as an operating loss, including the *multiclass cross-entropy*. Note that the Dice-approximating losses, such as the soft-Dice loss, can be adopted into the multilabel/multiclass modeling.

In literature, once the estimation is done, the predicted segmentation is produced by taking *argmax* or *thresholding* at 0.5 on the estimated probabilities. Indeed, *argmax* and *thresholding* are inspired by the decision-making in multiclass and multilabel classification, respectively. In segmentation, it is possible to attempt ad-hoc combinations, such as multiclass modeling + *thresholding*, and multilabel modeling + *argmax*. The major purpose of *argmax* is to provide *non-overlapping* prediction (e.g., the panoptic prediction in image segmentation). We discuss overlapping/non-overlapping segmentation in the next section.

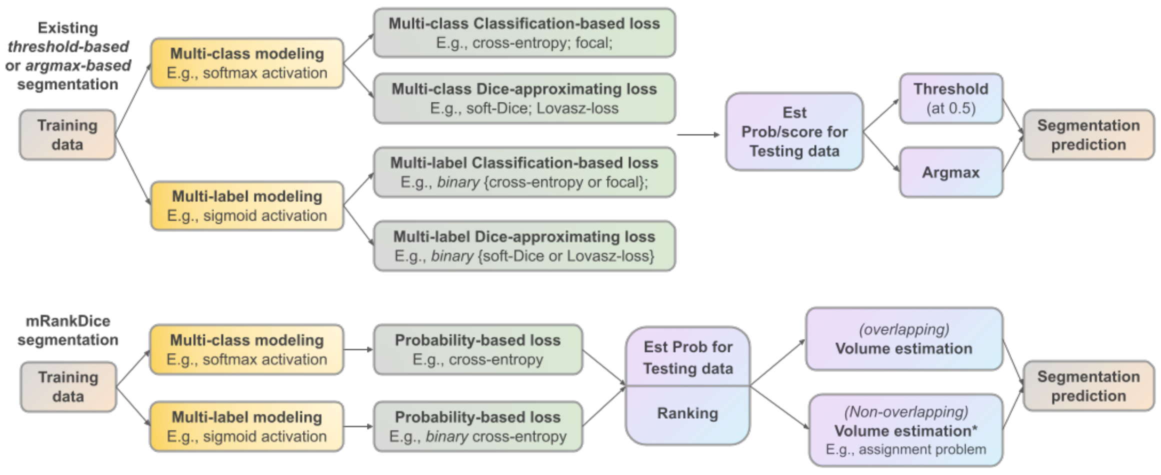
Existing *threshold-based* or *argmax-based* segmentation
Training data
Multi-class modeling E.g., softmax activation
Multi-label modeling E.g., sigmoid activation
Multi-class Classification-based loss E.g., cross-entropy; focal;
Multi-class Dice-approximating loss E.g., soft-Dice; Lovasz-loss
Multi-label Classification-based loss E.g., *binary* {cross-entropy or focal};
Multi-label Dice-approximating loss E.g., *binary* {soft-Dice or Lovasz-loss}
Est Prob/score for Testing data
Threshold (at 0.5)
Argmax
Segmentation prediction
mRankDice segmentation
Training data
Multi-class modeling E.g., softmax activation
Multi-label modeling E.g., sigmoid activation
Probability-based loss E.g., cross-entropy
Probability-based loss E.g., *binary* cross-entropy
Est Prob for Testing data
Ranking
(overlapping) Volume estimation
(Non-overlapping) Volume estimation* E.g., assignment problem
Segmentation prediction

Figure 2: The existing and the proposed frameworks under multiclass/multilabel modeling. The upper panel is *threshold-/argmax-based segmentation*, and the lower panel is the proposed *mRankDice* framework.

3.3 Overlapping or non-overlapping mDice-segmentation

Specifically, whether (or not) to allow for overlapping results among distinct classes leads to different decision-making procedures, namely overlapping/non-overlapping mDice-segmentation:

$$\text{(Overlapping)} \ \ \operatorname*{argmax}_{\boldsymbol{\Delta}} \ \mathrm{mDice}_\gamma(\boldsymbol{\Delta}), \quad \text{(Non-overlapping)} \ \ \operatorname*{argmax}_{\boldsymbol{\Delta}} \ \mathrm{mDice}_\gamma(\boldsymbol{\Delta}), \quad \sum_{k=1}^{K} \boldsymbol{\Delta}_k(\cdot) = \mathbf{1}_d. \tag{16}$$

In the overlapping setting, there is no restriction on a segmentation operator, thus the predicted segmentation for different classes may overlap. On the other hand, the overlapping is ruled out in non-overlapping formulation due the additional sum-to-one constraint. Lemma 6 presents the Bayes rule for overlapping segmentation, yielding that mDice-segmentation is reduced to class-specific Dice-segmentation subproblems if overlapping is allowed.

Lemma 6 (The Bayes rule for overlapping mDice-segmentation) *An overlapping (allowing) segmentation operator* $\boldsymbol{\Delta}^*$ *is a global maximizer of* $\mathrm{mDice}_\gamma(\boldsymbol{\Delta})$ *if and only if* $\boldsymbol{\Delta}^*_k$ *is the Bayes rule (global maximizer) of* $\mathrm{Dice}_{\gamma,k}(\boldsymbol{\Delta}_k)$ *on* $(\mathbf{X}, \mathbf{Y}_{\cdot k})$ *for all* $k \in \{1 \leq k \leq K : \alpha_k > 0\}$.

Therefore, it suffices to consider Dice-segmentation of each class separately in overlapping mDice-segmentation, and the proposed *RankDice* is readily extended to *mRankDice*; see Section 3.4.

Next, we investigate the Bayes rule for non-overlapping mDice-segmentation. To proceed, we denote $\boldsymbol{\Delta}^*$ as the Bayes rule (global maximizer) of non-overlapping mDice-segmentation in (16), and $\boldsymbol{\tau}^*(\cdot)$ as the volume function of the Bayes segmentation rule: $\boldsymbol{\tau}^*(\mathbf{x}) = (\tau_1^*(\mathbf{x}), \dots, \tau_K^*(\mathbf{x}))^\intercal = (\|\boldsymbol{\Delta}_1^*(\mathbf{x})\|_1, \cdots, \|\boldsymbol{\Delta}_K^*(\mathbf{x})\|_1)^\intercal$.

Lemma 7 *Suppose $\boldsymbol{\tau}^*(\cdot)$ is pre-known, then solving the Bayes rule for non-overlapping mDice-segmentation in* (16) *is equivalent to an assignment problem.*

As indicated in Lemma 7, even when the optimal volume function is pre-given, solving the Bayes rule for non-overlapping segmentation is nontrivial. For example, the most successful assignment algorithms, including the Hungarian method (Kuhn, 1955; Edmonds and Karp, 1972; Tomizawa, 1971) and its variants Jonker-Volgenant algorithm (Crouse, 2016), generally achieves an $O(d^3)$ running time complexity in our content, which is time-consuming for high-dimensional segmentation. In sharp contrast, for the overlapping case, when $\tau^*(\mathbf{x})$ is given, the Bayes rule is simply ranking all the conditional probabilities with $O(d\log(d))$. Moreover, when $\boldsymbol{\tau}^*(\mathbf{x})$ is unknown, the optimization for non-overlapping segmentation becomes a nonlinear integer optimization which is NP-hard in general (Murty and Kabadi, 1985; D'Ambrosio et al., 2020). Therefore, a fast $O(d\log(d))$ greedy approximation algorithm is more feasible in practical implementation. We leave pursuing this topic as future work.

Next, we summarize the proposed *mRankDice* framework under three scenarios.

3.4 mRankDice

In this section, we present the proposed *mRankDice* framework for mDice-segmentation. Before we proceed, we highlight the different roles of multiclass/multilabel modeling and the overlapping/non-overlapping constraint. Multiclass/multilabel modeling determines the probabilistic models (say the softmax or the sigmoid activation in a neural network) and an operating loss in probability estimation (say the cross-entropy or the binary cross-entropy). Meanwhile, the overlapping/non-overlapping constraint affects the segmentation prediction after the probabilities are estimated.

Therefore, we consider following segmentation three cases: "multilabel + overlapping", "multiclass + overlapping", and "multilabel/multiclass + non-overlapping".

mRankDice (multilabel + overlapping segmentation) is a straightforward extension of *RankDice* in Dice-segmentation (inspired by Lemma 6). Given a training dataset $(\mathbf{x}_i, \mathbf{y}_{i,1:d,1}, \cdots, \mathbf{y}_{i,1:d,K})_{i=1}^n$, with the same manner, the conditional probability function is estimated under a multilabel logistic regression (the binary cross-entropy loss):

$$\widehat{\mathbf{Q}}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{Q}\in\mathcal{Q}} \sum_{i=1}^n \sum_{j=1}^d \sum_{k=1}^K \Big(y_{ijk} \log\big(q_{jk}(\mathbf{x}_i)\big) + (1-y_{ijk}) \log\big(1-q_{jk}(\mathbf{x}_i)\big)\Big) + \lambda \|\mathbf{Q}\|^2, \qquad (17)$$

where $\mathbf{Q} = (q_{jk}) : \mathbb{R}^d \to [0,1]^{d\times K}$ is a matrix function, and $q_{jk}(\mathbf{x})$ is a candidate estimator of $p_{jk}(\mathbf{x})$. Then, given a new instance $\mathbf{x}$, the class-specific segmentation $\widehat{\boldsymbol{\Delta}}_k(\mathbf{x})$ is generated based on **Steps 2-3** in Section 2.2 with the estimated conditional probabilities $\widehat{\mathbf{Q}}_k(\mathbf{x})$ (the k-th column of $\widehat{\mathbf{Q}}(\mathbf{x})$). We summarize the foregoing computational scheme in Algorithm 2.

mRankDice (multiclass + overlapping segmentation) yields a different probability estimation procedure, where the conditional probability function is estimated under a multiclass logistic regression (the multiclass cross-entropy loss):

$$\widehat{\mathbf{Q}}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{Q}\in\mathcal{Q}} \sum_{i=1}^{n}\sum_{j=1}^{d}\sum_{k=1}^{K} y_{ijk}\log\big(q_{jk}(\mathbf{x}_i)\big) + \lambda\|\mathbf{Q}\|^2, \ \text{ s.t. } \sum_{k=1}^{K} q_{jk}(\cdot) = 1; \text{ for } j = 1,\cdots d, \tag{18}$$

where $\mathbf{Q} = (q_{jk}) : \mathbb{R}^d \to [0,1]^{d\times K}$ is a matrix function, and $\mathbf{q}_j(\mathbf{x})$ is a candidate estimator of $\mathbf{p}_j(\mathbf{x})$. Although the probability estimation (18) differs from (17), the downstream ranking and volume estimation remain the same according to Lemma 6. We also summarize the foregoing computational scheme in Algorithm 2.

Algorithm 2: mRankDice for overlapping mDice-segmentation.

```
   Input  : Training set: $(\mathbf{x}_i, \mathbf{y}_{i,1:d,1}, \cdots, \mathbf{y}_{i,1:d,K})_{i=1}^{n}$; A new testing instance: $\mathbf{x}$
   Output: The predicted segmentation for the testing instance $\widehat{\mathbf{\Delta}}(\mathbf{x})$
1  if multilabel outcome then
2      Multilabel Prob Est. Estimate the prob function $\widehat{\mathbf{Q}}$ via (17);
3  end
4  if multiclass outcome then
5      Multiclass Prob Est. Estimate the prob function $\widehat{\mathbf{Q}}$ via (18);
6  end
7  for $k = 1, \cdots, K$ do
8      Class-specific RankDice. Obtain $\widehat{\mathbf{\Delta}}_k(\mathbf{x})$ from Lines 2-22 in Algorithm 1 based on the
         estimated prob $\widehat{\mathbf{Q}}_k(\mathbf{x})$.
9  end
10 return The predicted segmentation $\widehat{\mathbf{\Delta}}(\mathbf{x}) = (\widehat{\mathbf{\Delta}}_1(\mathbf{x}), \cdots, \widehat{\mathbf{\Delta}}_K(\mathbf{x}))$
```

mRankDice (multiclass/multilabel + non-overlapping segmentation) is a quite difficult scenario for developing *mRankDice* from *RankDice* in Dice-segmentation. As indicated in Lemma 7, searching for an optimal non-overlapping mDice-segmentation operator is NP-hard in general. We leave pursuing this topic as future work.

4. Theory

In this section, we establish a theoretical foundation of segmentation, including the concept of Dice-calibration, the excess risk of the Dice metric, and the rate of convergence with respect to the excess risk of the proposed *RankDice* framework. For illustration, we focus on Dice-segmentation, and the results can be extended to mDice-segmentation, and *RankIoU* in terms of the IoU metric.

4.1 Dice-calibrated segmentation

In Theorem 1, the Bayes rule of Dice-segmentation is obtained. To carry this agenda further, we propose concept of "Dice-calibrated", following the same philosophy of Fisher consistency or classification-calibration in Lin (2004); Zhang (2004); Bartlett et al. (2006). Intuitively, Dice-calibration is the weakest possible condition on a consistent segmentation method with respect to the Dice metric, this is, at population level, a method ultimately searches for the Bayes rule that achieves the optimal Dice metric. Figure 3 indicates the overview and logical relations among the theoretic results in this section.

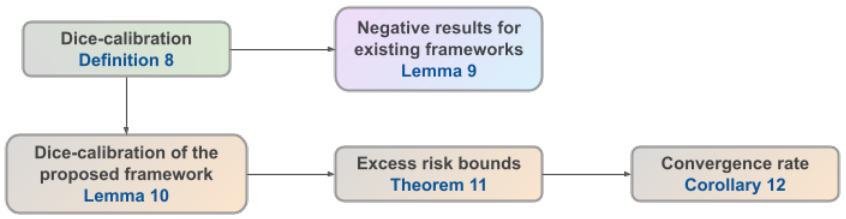

Figure 3: Flowchart of the theory for *RankDice* in Section 4.1, indicating the logical relations among the developed theorems.

To proceed, let $\mathcal{P}$ be the class of all probability measures $\mathbb{P}_{\mathbf{X},\mathbf{Y}}$ on (Borel) measurable subsets of $\mathbb{R}^d \times \{0,1\}^d$ such that $(\mathbf{X},\mathbf{Y}) \sim \mathbb{P}_{\mathbf{X},\mathbf{Y}}$, $(\mathbf{X},\mathbf{Y}) \in \mathbb{R}^d \times \{0,1\}^d$, and $Y_i \perp Y_j | \mathbf{X}$ for $i \neq j$. Denote $\mathcal{D}$ as the class of all (Borel) measurable segmentation operators $\boldsymbol{\delta} : \mathbf{x} \in \mathbb{R}^d \to \boldsymbol{\delta}(\mathbf{x}) \in \{0,1\}^d$. The definition of Dice-calibrated segmentation is given as follows.

Definition 8 (Dice-calibrated segmentation) *Given $\gamma \geq 0$, a (population) segmentation method $\mathcal{M}_\gamma : \mathcal{P} \to \mathcal{D}$ is Dice-calibrated if, for any $\mathbb{P}_{\mathbf{X},\mathbf{Y}} \in \mathcal{P}$,*

$$\mathrm{Dice}_\gamma\big(\mathcal{M}_\gamma(\mathbb{P}_{\mathbf{X},\mathbf{Y}})\big) = \mathrm{Dice}_\gamma(\boldsymbol{\delta}^*),$$

where $\boldsymbol{\delta}^*$ is the Bayes rule for Dice-segmentation defined in Theorem 1.

Now, we show that most existing loss functions, under the framework (2), are not Dice-calibrated.

Lemma 9 *Given a loss function $l(\cdot,\cdot)$, define $\mathcal{M}_\gamma(\mathbb{P}_{\mathbf{X},\mathbf{Y}})$ under the framework (2), that is,*

$$\mathcal{M}_\gamma(\mathbb{P}_{\mathbf{X},\mathbf{Y}})(\mathbf{x}) := \mathbf{1}(\widetilde{\mathbf{q}}_l(\mathbf{x}) \geq 0.5), \quad \widetilde{\mathbf{q}}_l = \underset{\mathbf{q}}{\mathrm{argmin}}\ \mathbb{E}\big(l(\mathbf{Y},\mathbf{q}(\mathbf{X}))\big).$$

Then, $\mathcal{M}_\gamma(\mathbb{P}_{\mathbf{X},\mathbf{Y}})$ is not Dice-calibrated for $\gamma = 0$ when $l(\cdot,\cdot)$ is the soft-Dice loss or any classification-calibrated loss, including the cross-entropy loss and the focal loss.

Lemma 9 indicates that the existing framework (2) with most losses ultimately yields a sub-optimal solution to Dice-segmentation, even if a "classification-calibrated" loss, such as the cross-entropy loss or the focal loss (Charoenphakdee et al., 2021), is used. Meanwhile, the result for the soft-Dice loss in Lemma 9 is in line with the empirical results in Bertels et al. (2019), where optimization with the soft-Dice loss can introduce a volumetric bias for tasks with high inherent uncertainty. In sharp contrast, the proposed *RankDice* method is Dice-calibrated (Lemma 10) and its asymptotic convergence rate in terms of the Dice metric is provided in Theorem 11.

To proceed, we give the definition of *RankDice* at population level in Appendix B.1, which replaces the average in (5) by the population expectation. Moreover, the cross-entropy loss in (5) can be extended to an arbitrary *strictly proper* loss (Gneiting and Raftery, 2007). The most common strictly proper losses are the cross-entropy loss and the squared error loss.

Lemma 10 (Dice-calibrated) *The proposed RankDice framework with a strictly proper loss is Dice-calibrated.*

Next, we present an excess risk bound in terms of the Dice metric, that is, $\text{Dice}(\boldsymbol{\delta}^*) - \text{Dice}(\widehat{\boldsymbol{\delta}})$.

Theorem 11 (Excess risk bounds) *Given $\gamma \geq 0$, let $\widehat{\mathbf{q}}(\cdot)$ be an estimated probability of $\mathbf{p}(\cdot)$, and $\widehat{\boldsymbol{\delta}}(\cdot)$ be the RankDice segmentation function defined in* (7) *based on $\widehat{\mathbf{q}}(\cdot)$, then*

$$\text{Dice}_\gamma(\boldsymbol{\delta}^*) - \text{Dice}_\gamma(\widehat{\boldsymbol{\delta}}) \leq C_1 \mathbb{E}_{\mathbf{X}} \|\widehat{\mathbf{q}}(\mathbf{X}) - \mathbf{p}(\mathbf{X})\|_1, \tag{19}$$

where $C_1 > 0$ is a universal constant depending only on γ.

As indicated in Theorem 11, the excess risk of the Dice metric for the proposed *RankDice* framework is upper bounded by the total variation (TV) distance between the estimated probability $\widehat{\mathbf{q}}$ and the true probability $\mathbf{p}$. Note that the Kullback-Leibler divergence (the excess risk for the cross-entropy) dominates the TV distance. It follows that if the KL divergence between p_j and $\widehat{q}_j$ goes to 0, then $\widehat{\mathbf{q}}$ converges to $\mathbf{p}$ in the TV sense, and so does $\text{Dice}_\gamma(\widehat{\boldsymbol{\delta}})$ to $\text{Dice}_\gamma(\boldsymbol{\delta}^*)$.

Taken together, we present the rate of convergence for the empirical estimator obtained from the proposed *RankDice* framework (Steps 1-3) in Section 2.2.

Corollary 12 (Convergence rate) *Let $\widehat{\mathbf{q}}(\cdot)$ and $\widehat{\boldsymbol{\delta}}(\cdot)$ be obtained by the proposed RankDice framework (Steps 1-3) in Section 2.2, and*

$$\mathcal{E}_{CE}(\widehat{\mathbf{q}}) := \mathbb{E}\Big(l_{CE}\big(\mathbf{Y}, \widehat{\mathbf{q}}(\mathbf{X})\big)\Big) - \mathbb{E}\Big(l_{CE}\big(\mathbf{Y}, \mathbf{p}(\mathbf{X})\big)\Big) = O_P(\varepsilon_n),$$

where $l_{CE}(\cdot,\cdot)$ is defined in (3)*. Then,*

$$\text{Dice}_\gamma(\boldsymbol{\delta}^*) - \text{Dice}_\gamma(\widehat{\boldsymbol{\delta}}) = O_P(\sqrt{d}\varepsilon_n^{1/2}). \tag{20}$$

Note that $\mathcal{E}_{\text{CE}}$ is the excess risk of the cross-entropy loss or the negative conditional log-likelihood in (5), and its asymptotics as well as a rate of convergence can be established based on statistical learning theory of empirical risk minimization (Pollard, 1984; Shen, 1997; Bartlett et al., 2005; Giné and Koltchinskii, 2006; Cucker and Zhou, 2007), which depends on the sample size and the complexity of the probability class. Then, the rate of convergence of the excess risk in terms of the Dice metric is obtained via (20). Note that both (19) and (20) are derived for a fixed dimension, and the upper bounds can be extended and improved when the dimension of segmentation grows with the sample size.

Finally, we briefly discuss the connections of the developed theory (i.e., Lemma 10, Theorem 11 and Corollary 12) with the existing results. For example, Popordanoska et al. (2021) derived an upper bound for the volume bias $\|\mathbb{E}_{\mathbf{X}}(\widehat{\mathbf{q}}(\mathbf{X}) - \mathbf{p}(\mathbf{X}))\|_1$ in terms of the TV distance. It is worth noting that the volume bias focuses on conditional probability estimation and a small volume bias may not necessarily yield a consistent segmentation rule in terms of the Dice metric. In contrast, our result on the excess risk $\text{Dice}(\boldsymbol{\delta}^*) - \text{Dice}(\widehat{\boldsymbol{\delta}})$ characterizes the performance of segmentation rule $\widehat{\boldsymbol{\delta}}$. Besides, Bao and Sugiyama (2020) proved the consistency of their method under a linear fractional approximation of Dice metric (see Appendix A), which seems not directly comparable to ours.

4.2 Relation between Dice and IoU metrics

In this section, we consider the relation and difference between Dice and IoU metrics, and present the Bayes rule for IoU-segmentation.

Lemma 13 *A segmentation rule $\boldsymbol{\delta}^*$ is a global maximizer of* $\mathrm{IoU}_\gamma(\boldsymbol{\delta})$ *if and only if it satisfies that*

$$\delta_j^*(\mathbf{x}) = \begin{cases} 1 & \text{if } p_j(\mathbf{x}) \text{ ranks top } \tau^*(\mathbf{x}), \\ 0 & \text{otherwise}. \end{cases}$$

The optimal volume $\tau^*(\mathbf{x})$ is given as

$$\tau^*(\mathbf{x}) = \operatorname*{argmax}_{\tau \in \{0,1,\cdots,d\}} \Big(\sum_{j \in J_\tau(\mathbf{x})} p_j(\mathbf{x}) + \gamma \Big) \sum_{l=0}^{d-\tau} \frac{1}{\tau + l + \gamma} \mathbb{P}\big(\Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = l\big), \tag{21}$$

where $J_\tau(\mathbf{x}) = \big\{ j : \sum_{j'=1}^{d} \mathbb{I}\big(p_{j'}(\mathbf{x}) \geq p_j(\mathbf{x})\big) \leq \tau \big\}$ is the index set of the τ-largest conditional probabilities with $J_0(\mathbf{x}) = \emptyset$, and $\Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = \sum_{j' \notin J_\tau(\mathbf{x})} B_{j'}(\mathbf{x})$ is a Poisson-binomial random variable, and $B_j(\mathbf{x})$ is a Bernoulli random variable with the success probability $p_j(\mathbf{x})$.

In view of Lemma 13, IoU-segmentation shares a substantial similarity with Dice-segmentation in terms of the Bayes rule. On this ground, a consistent *RankIoU* framework is also developed based on a *plug-in* rule by replacing $\mathbf{p}(\mathbf{x})$ as $\widehat{\mathbf{q}}(\mathbf{x})$. Specifically, *RankIoU* comprises three steps, where **Steps 1-2** are the same as in *RankDice*; see Section 2.2.
Step 3′ (IoU volume estimation): From (4), we estimate the volume $\widehat{\tau}(\mathbf{x})$ by replacing the true conditional probability $\mathbf{p}(\mathbf{x})$ with the estimated one $\widehat{\mathbf{q}}(\mathbf{x})$:

$$\widehat{\tau}(\mathbf{x}) = \operatorname*{argmax}_{\tau \in \{0,1,\cdots,d\}} \Big(\sum_{j \in J_\tau(\mathbf{x})} \widehat{q}_j(\mathbf{x}) + \gamma \Big) \sum_{l=0}^{d-\tau} \frac{1}{\tau + l + \gamma} \mathbb{P}\big(\widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = l\big),$$

where $\widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = \sum_{j \notin J_\tau(\mathbf{x})} \widehat{B}_j(\mathbf{x})$ is a Poisson-binomial random variable, and $\widehat{B}_j(\mathbf{x})$ are independent Bernoulli random variables with success probabilities $\widehat{q}_j(\mathbf{x})$; for $j = 1, \cdots, d$.

Similar to *RankDice*, the predicted IoU-segmentation $\widehat{\boldsymbol{\delta}}(\mathbf{x}) = (\widehat{\delta}_1(\mathbf{x}), \cdots, \widehat{\delta}_d(\mathbf{x}))^\intercal$ is produced by taking the top-$\widehat{\tau}(\mathbf{x})$ conditional probabilities:

$$\widehat{\delta}_j(\mathbf{x}) = 1, \text{ if } j \in \{j_1, \cdots, j_{\widehat{\tau}(\mathbf{x})}\}; \quad \widehat{\delta}_j(\mathbf{x}) = 0, \text{ otherwise.}$$

For multiclass/multilabel segmentation, the conditional probability estimation (17) and (18) are carried over into *mRankIoU* and the subsequent ranking and volume estimation remain the same as **Step 2** and **Step 3′** in binary segmentation.

Computationally, *RankIoU* involves the evaluation of $\mathbb{P}\big(\widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = l\big)$ in **Step 3′**. The FFT algorithm and the truncated refined normal approximation (T-RNA) are applicable after minor modifications; however, the blind approximation (BA) may not be appropriate due to the discrepancy of $\widehat{\Gamma}(\mathbf{x})$ and $\widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})$, especially when the size of $J_\tau(\mathbf{x})$ is large; see Section 2.3. Thus, the computation scheme of *RankIoU* might be relatively expensive in high-dimensional segmentation. Here, we present a parallel result of Lemma 3 to narrow down the searching range in **Step 3′** of *RankIoU*.

Lemma 14 *If*

$$\sum_{s=1}^{\tau} \widehat{q}_{j_s}(\mathbf{x}) + \gamma \geq \frac{\widehat{q}_{j_{\tau+1}}(\mathbf{x})}{1-\widehat{q}_{j_{\tau+1}}(\mathbf{x})} \max\Big(d+\gamma, \frac{((d-\tau)\widehat{q}_{j_{\tau+1}}(\mathbf{x})+\tau+\gamma)^2}{\tau+\gamma}\Big),$$

then $\varpi_\tau(\mathbf{x}) \geq \varpi_{\tau'}(\mathbf{x})$ for all $\tau' > \tau$, where

$$\varpi_\tau(\mathbf{x}) = \Big(\sum_{j\in J_\tau(\mathbf{x})} \widehat{q}_j(\mathbf{x}) + \gamma\Big) \sum_{l=0}^{d-\tau} \frac{1}{\tau+l+\gamma} \mathbb{P}\big(\widehat{\Gamma}_{-J_\tau(\mathbf{x})}(\mathbf{x}) = l\big).$$

Theoretically, the concept of "IoU-calibrated" can be established (by replacing Dice as IoU in Definition 8) and the excess risk bounds can be derived in parallel to Dice-segmentation.

Theorem 15 *Given $\gamma \geq 0$, let $\widehat{\mathbf{q}}(\cdot)$ be an estimated probability of $\mathbf{p}(\cdot)$, and $\widehat{\boldsymbol{\delta}}(\cdot)$ be the RankIoU segmentation function based on $\widehat{\mathbf{q}}(\cdot)$, then*

$$\mathrm{IoU}_\gamma(\boldsymbol{\delta}^*) - \mathrm{IoU}_\gamma(\widehat{\boldsymbol{\delta}}) \leq C_2 \mathbb{E}_{\mathbf{X}} \|\widehat{\mathbf{q}}(\mathbf{X}) - \mathbf{p}(\mathbf{X})\|_1,$$

where $C_2 > 0$ is a universal constant depending only on γ. Consequently, if $\mathcal{E}_{CE}(\widehat{\mathbf{q}}) = O_P(\varepsilon_n)$, then

$$\mathrm{IoU}_\gamma(\boldsymbol{\delta}^*) - \mathrm{IoU}_\gamma(\widehat{\boldsymbol{\delta}}) = O_P(\sqrt{d}\varepsilon_n^{1/2}).$$

5. Numerical experiments

This section describes a set of simulations and real datasets that demonstrate the segmentation performance of the proposed *RankDice* and *mRankDice* frameworks compared with the existing *argmax*- and *thresholding*-based frameworks using various loss functions and network architectures. For illustration, the segmentation performances for all numerical experiments are evaluated by empirical Dice/IoU metrics with $\gamma = 0$, see Appendix A. For the mDice/mIoU metric, the class-specific weight is defined as in (15). All experiments are conducted using PyTorch and CUDA on an NVIDIA GeForce RTX 3080 GPU. All Python codes are available for download at our GitHub repository (`https://github.com/statmlben/rankseg`).

5.1 Simulation

In this section, we mainly compare the proposed *RankDice* framework with the *thresholding*-based framework (2) in various simulated examples. Note that for Dice-segmentation with binary outcomes, *threshold*- and *argmax*-based frameworks yield the same solution. Both frameworks require an estimation of conditional probability function in the first stage. Therefore, in order to convincingly demonstrate the difference between two frameworks, in our simulation, we assume the true conditional probabilities $p_j(\mathbf{x}) = \mathbb{P}(Y_j|\mathbf{x}); j = 1, \cdots, d$ are perfectly estimated, and report the Dice metric of the downstream segmentation produced by two different frameworks.

Example 1. To mimic the spatial smoothness in practical segmentation problem, especially for image segmentation, the simulated dataset based on matrix response ($d = W \times H$) is generated as follows. First, the true probability matrix $\mathbf{P} = (p_{wh})_{W\times H}$ are generated by two patterns:

- Step decay: $p_{wh} \sim U(0.5, 1)$, if $w \leq \lfloor \rho W \rfloor$ and $h \leq \lfloor \rho H \rfloor$; $p_{wh} \sim N_{[0,1]}(\beta, 0.1)$, otherwise.

- Exponential decay: $p_{wh} = \exp\big(-\beta(w+h)\big)$, as visualized in the upper panel of Figure 6.
- Linear decay: $p_{wh} = 1-\beta(w+h)/(W+H)$, as visualized in the lower panel of Figure 6.

Here β is a decay parameter, $U(0,1)$ is the uniform distribution, and $N_{[0,1]}(\beta,0.1)$ is the truncated normal distribution with mean β and standard deviation 0.1. In our simulation, we consider $\beta = 0.1, 0.3, 0.5$ with $\rho = 0.1$ for step decay, $\beta = 1.01, 1.05, 1.10$ for exponential decay, and $\beta = 1,2,4$ for linear decay. For each case, four different dimensions are considered: $W = H = 28, 64, 128, 256$, and therefore d increases from 784 to 65,536. Then, the proposed *RankDice* framework and the *thresholding*-based framework (at 0.5) are conducted on the true probability matrix $\mathbf{P}$. Both decay scenarios are replicated 100 times, and the averaged Dice metrics and its standard errors are summarized in Table 9.

Example 2. As indicated in Theorem 1 and Remark 2, the optimal segmentation volume (or the optimal threshold) varies significantly across different inputs. This example aims to illustrate the *suboptimality of (tuning a threshold of) a fixed thresholding* based on step decay in Example 1. To this end, the conditional probability matrices $(\mathbf{P}_i)_{i=1,\cdots,n}$ of inputs are generated as follows. First, $\rho_i \sim U(0,1)$ to mimic the different segmentation scales/patterns over images in real applications. Next, $\mathbf{P}_i$ is generated by step decay based on $\beta = 0.1$ and $\rho = \rho_i$. Then, the proposed *RankDice* framework and the fixed *thresholding*-based framework (at 0.1. $\cdots$, 0.9) are applied as the same manner in Example 1. All methods are applied with $n = 2000$ and $W = H = 64$, and the averaged Dice metrics and its standard errors are summarized in Table 10. In this example, the heterogeneous conditional probabilities yield different optimal segmentation volumes (or thresholds) for images, thus (tuning a threshold on) a fixed thresholding leads to a suboptimal solution. To better illustrate the adaptiveness over optimal thresholds, we also present the optimal thresholds for different images with various generating parameters (β,ρ) on segmentation patterns of Example 2 in Figure 7, and similar results are demonstrated in Figure 4 for Pascal VOC 2012 dataset.

The major conclusions on the simulated examples are listed as follows.

- It is evident that RankDice significantly outperforms the *thresholding*-based (at 0.5) framework in both decay scenarios with various dimensions (Table 9). The substantial improvement is consistent with the findings of Lemma 1, which indicates that segmentation and classification are entirely distinct problems.
- Interestingly, the amount of improvement is gradually increased when the decay of probabilities becomes progressively faster (for exponential decay and linear decay). This suggests that the proposed *RankDice* might be even more advantageous in *well-separated segmentation*.
- As suggested in Table 10, the proposed *RankDice* generally outperforms a fixed thresholding framework (with any threshold). This because that the optimal threshold can vary greatly across inputs, as indicated in Figure 7. Moreover, tuning the threshold may improve the performance of the thresholding-based framework, yet it still leads to a suboptimal solution compared with the proposed *RankDice*.

5.2 Real datasets

This section examines the performance of the proposed *RankDice* framework in the PASCAL VOC 2012 (Everingham et al., 2012), the fine-annotated CityScapes semantic segmentation benchmark

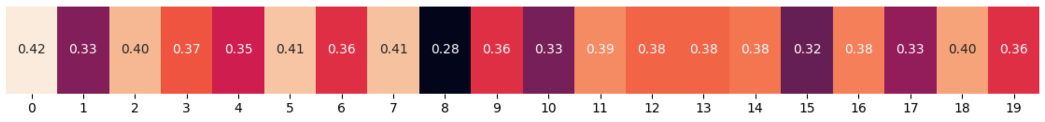

Figure 4: The heatmap for averaged (over all validation samples) optimal thresholding probabilities (i.e. the probability cutpoint $\widehat{q}_{j_{\widehat{\tau}}}$) provided by the proposed *RankDice*, see Remark 2 for more details. Here, the x-axis indicates the class, and the results are provided by a PSPNet trained with the cross-entropy loss.

(Cordts et al., 2016), and Kvasir SEG dataset polyp segmentation dataset (Ajtai et al., 1983). Three different neural network architectures are considered: DeepLab-V3+ (Chen et al., 2018) with resnet101 as the backbone, PSPNet with resnet50 as the backbone, and FCN8 with resnet101 as the backbone. We report both mDice and mIoU metrics of the segmentation produced by *Threshold*, *Argmax* and *RankDice* on the *same* estimated network/probability. Note that only overlapping segmentation is considered.

Fine-annotated Cityscapes dataset contains 5,000 high quality pixel-level annotated images. For all methods, we employ SGD on the learning rate (`lr`) schedule `lr_schedule='poly'`, and the initial learning rate `initial_lr=0.01`, `weight_decay=100`, `momentum=0.9`, crop size 512x512, batch size 6, and 300 epochs. The performance on validation set is measured in terms of the mDice and mIoU averaged across 19 object classes (Table 2).

Pascal VOC 2012 dataset contains 20 foreground object classes and one background class. The dataset contains 1,464 training and 1,449 validation pixel-level annotated images. We augment the dataset by using the additional annotations provided by Hariharan et al. (2011). For all methods, we employ SGD on `lr_schedule='poly'`, and the initial learning rate `initial_lr=0.01`, `weight_decay=100`, `momentum=0.9`, crop size 480x480, batch size 8, and an early stop with patient 10 based on validation loss. The performance on validation set is measured in terms of the mDice and mIoU averaged across the 20 object classes (Table 3). In this dataset, we also present a heatmap (Figure 4) for averaged minimal estimated probabilities for segmented features (i.e. $\widehat{q}_{j_{\widehat{\tau}}}$) by the proposed *RankDice*, to highlight its difference to the *thresholding*-based framework.

Kvasir SEG dataset contains 1000 polyp images and their ground truth segmentation (a single class) from the Kvasir dataset. The scale of the images varies from 332x487 to 1920x1072 pixels. For all methods, we employ SGD on `lr_schedule='poly'`, and the initial learning rate `initial_lr=0.01`, `weight_decay=100`, `momentum=0.9`, crop size 320x320, batch size 8, and 140 epochs. The performance on testing set is measured in terms of the Dice and IoU (Table 4).

Multiclass/multilabel loss. In both datasets, we use six loss functions (including multiclass and multilabel losses) in the implementation, including the cross-entropy (CE), the focal loss (Focal), the binary cross-entropy (BCE), the soft-Dice loss (Soft-Dice), the binary soft-Dice loss (B-Soft-Dice), and the LovaszSoftmax loss (LovaszSoftmax). For multiclass losses, including CE, Focal, Soft-Dice, and LovaszSoftmax, we use the softmax function as the output layer activation function of a neural network. For multilabel losses, including BCE and B-Soft-Dice, we use the sigmoid function as the output layer activation function of a neural network.

Dice-segmentation based on a single class. For the first two datasets, when we focus on a single object class, it reduces to a Dice-segmentation with binary outcomes. To examine the performance for the proposed *RankDice* in binary segmentation, we also report the Dice/IoU metric for each label separately. In principle, we need to train a model only on the binary label for an object class, then produce the segmentation prediction by *thresholding* (or *argmax*) and *RankDice*. However, based on our empirical study, a model trained from full labels is significantly better than the one trained from a single binary label. We thus train a model based the same procedure in the aforementioned segmentation with multiclass/multilabel losses on full labels, and then produce the prediction based on the estimated probability for each object class separately. The best two performances ("PSPNet + CE" and "PSPNet + BCE") for both datasets are summarized in Tables 5 and 6. The performance for other models and losses can be found in the supplementary. In Figure 5, we present the segmentation results on illustrative examples for all methods to demonstrate the difference between the proposed *RankDice* and the existing methods.

The fixed-thresholding framework with different thresholds. As indicated in Remark 2 and Example 2 in Section 5.1, the optimal segmentation volume (or the optimal thresholding) varies significantly across different images, thus (tuning on) a fixed-thresholding yields a suboptimal solution. In Tables 7 and 10, we also report the numerical performance based on different thresholds for real datasets and Example 2, respectively.

Probability calibration via Temperature scaling (TS). According to Theorem 11, the consistency of the proposed method holds if the estimated conditional probabilities are calibrated. Alternatively, improving the probability calibration may improve the segmentation performance for the proposed framework. Therefore, in Table 8, we examine the numerical results of the proposed method via TS (with different tuning temperatures), which is one of the most effective probability calibration methods as suggested by Guo et al. (2017).

Model	Loss	Threshold (at 0.5) (mDice, mIoU) ($\times$.01)	Argmax (mDice, mIoU) ($\times$.01)	mRankDice (our) (mDice, mIoU) ($\times$.01)
DeepLab-V3+ (resnet101)	CE	(56.00, 48.40)	(54.20, 46.60)	(**57.80**, **49.80**)
	Focal	(54.10, 46.60)	(53.30, 45.60)	(56.50, 48.70)
	BCE	(49.80, 24.90)	(44.20, 22.10)	(54.00, 27.00)
	Soft-Dice	(39.50, 35.90)	(39.50, 35.90)	(39.50, 35.90)
	B-Soft-Dice	(41.00, 20.50)	(27.60, 13.80)	(41.10, 20.50)
	LovaszSoftmax	(55.20, 47.60)	(52.30, 45.10)	(55.50, 47.80)
PSPNet (resnet50)	CE	(57.50, 49.60)	(56.50, 48.50)	(**59.30**, **51.00**)
	Focal	(56.00, 48.20)	(55.80, 47.70)	(58.20, 50.00)
	BCE	(51.40, 25.70)	(47.60, 23.80)	(55.10, 27.60)
	Soft-Dice	(49.10, 43.50)	(48.70, 43.20)	(49.30, 43.60)
	B-Soft-Dice	(46.30, 23.10)	(32.70, 16.40)	(46.20, 23.10)
	LovaszSoftmax	(56.80, 48.90)	(55.40, 47.70)	(56.70, 49.10)
FCN8 (resnet101)	CE	(51.40, 43.70)	(50.50, 42.60)	(**53.50**, **45.30**)
	Focal	(48.50, 41.20)	(49.60, 41.60)	(51.50, 43.70)
	BCE	(39.40, 19.70)	(39.40, 19.70)	(41.30, 20.60)
	Soft-Dice	(28.30, 24.30)	(28.30, 24.30)	(28.30, 24.80)
	B-Soft-Dice	(29.10, 14.60)	(29.10, 14.60)	(29.10, 14.60)
	LovaszSoftmax	(48.10, 40.40)	(42.90, 35.80)	(48.90, 40.90)

Table 2: Averaged mDice and mIoU metrics of *Threshold*, *Argmax*, and the proposed *mRankDice* based on state-of-the-art models/losses on **Fine-annotated CityScapes** *val* set. Gray color indicates that *RankDice/mRankDice* is inappropriately applied to a loss function which is not strictly proper. The best performance in each model is bold-faced.

Overall, the empirical results show that the proposed *RankDice/mRankDice* framework yields good performance in three segmentation benchmarks. The major empirical conclusions on the proposed RankDice are listed as follows.

Model	Loss	Threshold (at 0.5) (mDice, mIoU) ($\times$.01)	Argmax (mDice, mIoU) ($\times$.01)	mRankDice (our) (mDice, mIoU) ($\times$.01)
DeepLab-V3+ (resnet101)	CE	(63.60, 56.70)	(61.90, 55.30)	(64.01, 57.01)
	Focal	(62.70, 55.01)	(60.50, 53.20)	(62.90, 55.10)
	BCE	(63.30, 31.70)	(59.90, 29.90)	(**64.60**, **32.30**)
	Soft-Dice	—	—	—
	B-Soft-Dice	—	—	—
	LovaszSoftmax	(57.70, 51.60)	(56.20, 50.30)	(57.80, 51.60)
PSPNet (resnet50)	CE	(64.60, 57.10)	(63.20, 55.90)	(65.40, 57.80)
	Focal	(64.00, 56.10)	(63.90, 56.10)	(66.60, 58.50)
	BCE	(64.20, 32.10)	(65.20, 32.60)	(**67.10, 33.50**)
	Soft-Dice	(59.60, 54.00)	(58.80, 53.20)	(60.00, 54.30)
	B-Soft-Dice	(63.30, 31.60)	(54.00. 27.00)	(64.30, 32.20)
	LovaszSoftmax	(62.00, 55.20)	(60.80, 54.10)	(62.20, 55.40)
FCN8 (resnet101)	CE	(49.50, 41.90)	(45.30, 38.40)	(50.40, 42.70)
	Focal	(50.40, 41.80)	(47.20, 39.30)	(**51.50, 42.50**)
	BCE	(46.20, 23.10)	(44.20, 22.10)	(47.70, 23.80)
	Soft-Dice	—	—	—
	B-Soft-Dice	—	—	—
	LovaszSoftmax	(39.80, 34.30)	(37.30, 32.20)	(40.00, 34.40)

Table 3: Averaged mDice and mIoU of *threshold*, *argmax*, and the proposed *mRankDice* based on state-of-the-art models/losses on **PASCAL VOC 2012** *val* set. "—" indicates that either the performance is significantly worse or the training is unstable. Gray color indicates that *RankDice/mRankDice* is inappropriately applied to a loss function which is not strictly proper. The best performance in each model is bold-faced.

Model	Loss	Threshold/Argmax (Dice, IoU) ($\times$.01)	mRankDice (our) (Dice, IoU) ($\times$.01)
DeepLab-V3+ (resnet101)	CE	(87.9, 80.7)	(**88.3**, **80.9**)
	Focal	(86.5, 87.3)	(83.1, 73.2)
	Soft-Dice	(85.7, 77.8)	(85.8, 77.9)
	LovaszSoftmax	(84.3, 77.3)	(84.5, 77.4)
PSPNet (resnet50)	CE	(86.3, 79.2)	(**87.1**, **79.8**)
	Focal	(83.8, 75.4)	(81.8, 72.4)
	Soft-Dice	(83.5, 75.9)	(83.7, 76.1)
	LovaszSoftmax	(86.0, 79.2)	(86.0, 79.2)
FCN8 (resnet101)	CE	(81.9, 73.5)	(**82.1**, **73.6**)
	Focal	(78.5, 69.0)	(**70.3, 58.3**)
	Soft-Dice	—	—
	LovaszSoftmax	(82.0, 73.4)	(82.0, 73.4)

Table 4: Averaged mDice and mIoU of *threshold/argmax*, and the proposed *mRankDice* based on state-of-the-art models/losses on **Kvasir SEG** dataset (with a single class segmentation). "—" indicates that either the performance is significantly worse. Gray color indicates that *RankDice* is inappropriately applied to a loss function which is not strictly proper. The best performance in each model is bold-faced.

- As suggested in Tables 2 and 3, the proposed RankDice framework *consistently* outperforms the *threshold*-based and *argmax*-based frameworks based on the same trained model/network. The percentage of improvement on the best performance (for each framework) are 3.13% (over *threshold*) and 4.96% (over *argmax*) for CityScapes dataset (PSPNet + CE), and 3.87% (over *threshold*) and 2.91% (over *argmax*) for Pascal VOC 2012 dataset (PSPNet + CE/BCE), and 0.926% for Kvasir SEG dataset (PSPNet + CE).

- For Dice-segmentation based on a single class, as suggested in Tables 5 and 6, the proposed *RankDice* framework *consistently* outperforms the *threshold-/argmax*-based framework for each class. The largest percentage of class-specific improvement in terms of the Dice metric on the best performance (for each framework) is 23.6% for CityScapes dataset, and 26.9% for Pascal VOC 2021 dataset.

- The proposed *RankDice* works significantly better in "difficult" segmentation. As indicated in Tables 5 and 6, the improvements by *RankDice* are negatively correlated with the resulting Dice/IoU metrics. It is also suggested by Figure 5, where we illustrate three images from classes `cat` (no improvement) and `chair` (26.9% improvement). As presented in all examples of `chair` and the last example of `cat`, the improvement is significant when segmentation is difficult (the target object is either occluded or similar with other objects).

- As suggested in Table 7, the empirical results are in line with Remark 2 (theoretically) and Table 10 (numerically), suggesting that the proposed *RankDice* generally outperforms a fixed thresholding framework (with any threshold). This because that the optimal threshold can vary greatly across images, as indicated in Figure 4. Moreover, tuning the threshold may improve the performance of the fixed-thresholding framework, yet it still leads to a suboptimal solution compared with the proposed *RankDice*.

- As suggested in Table 8, the segmentation performance can be potentially improved by TS probability calibration method, especially 4.59% improvement for CE loss and 3.28% improvement for BCE in Pascal VOC 2021 dataset. Yet, the TS demands an additional validation dataset to tune the optimal temperature, thus more numerical experiments are required to suggest the effectiveness of this promising method. We leave pursuing this topic as future work.

- As expected, the proposed *RankDice/mRankDice* performs well for strictly proper loss functions, including CE and BCE. In addition, we show that the performance is continuously improved compared with existing frameworks for some classification calibrated (only) losses, such as the focal loss. It is possible that this phenomenon is due to the relationship between the estimated scores (from focal loss) and the true conditional probabilities (cf. Charoenphakdee et al. (2021); Liu et al. (2021)). We leave this topic as future work.

- Although the *RankDice/mRankDice* framework is developed for Dice/mDice optimization, the performance in terms of the IoU/mIoU metric is also consistently improved.

Moreover, we also present some important observations based on our experiments about losses, frameworks, and models.

- CE, Focal and BCE are the top three losses for Dice-segmentation. While some Dice approximating losses, such as Soft-Dice and binary Soft-Dice, usually lead to suboptimal solutions.

- It seems that the multiclass modeling and multiclass losses are more preferred for both datasets. Moreover, the *threshold*-based framework usually outperforms the *argmax*-based framework for both multiclass and multilabel losses.

- For multiclass losses, including CE Focal, Soft-Dice, and LovaszSoftmax, the Dice and IoU metrics are consistent, i.e., a higher Dice yields a higher IoU score. For multilabel losses, including BCE and B-Soft-Dice, there is a significant difference between Dice and IoU metrics.

Object class	Threshold/Argmax (Dice, IoU) ($\times .01$)			RankDice (our) (Dice, IoU) ($\times .01$)			**imps.** (best vs. best)
	CE	Focal	BCE	CE	Focal	BCE	(Dice)
`road`	(85.7, 77.2)	(86.1, 77.9)	(92.2, 46.1)	(85.6, 77.1)	(86.0, 77.7)	(92.2, 46.1)	$\sim$
`sidewalk`	(57.3, 47.8)	(53.6, 43.8)	(43.8, 21.9)	(60.8, 50.8)	(58.7, 48.4)	(54.0, 27.0)	6.1%
`building`	(84.6, 76.2)	(83.4, 74.8)	(79.4, 39.7)	(85.1, 76.7)	(83.6, 74.7)	(82.1, 41.0)	$\sim$
`wall`	(17.4, 13.6)	(16.1, 12.4)	(04.9, 02.4)	(21.0, 16.4)	(21.5, 16.8)	(08.3, 04.2)	23.6%
`fence`	(14.7, 10.9)	(12.4, 08.9)	(15.2, 07.6)	(15.8, 11.7)	(13.7, 09.8)	(19.1, 09.5)	25.7%
`pole`	(41.9, 29.0)	(34.7, 23.4)	(27.1, 13.5)	(46.0, 31.7)	(35.6, 23.1)	(36.4, 18.2)	9.8%
`traffic light`	(34.9, 26.5)	(31.5, 24.0)	(18.7, 09.4)	(37.4, 28.3)	(33.5, 24.7)	(21.3, 10.6)	7.2%
`traffic sign`	(49.9, 39.0)	(45.9, 35.1)	(35.3, 17.6)	(51.4, 40.1)	(46.6, 35.1)	(39.6, 19.8)	3.0%
`vegetation`	(90.2, 84.1)	(90.2, 83.8)	(89.0, 44.5)	(90.3, 84.1)	(89.6, 82.8)	(89.4, 44.7)	$\sim$
`terrain`	(25.7, 20.1)	(24.1, 18.5)	(19.8, 09.9)	(29.4, 23.1)	(28.7, 22.7)	(25.3, 12.7)	14.4%
`sky`	(83.6, 77.0)	(82.0, 75.2)	(80.1, 40.0)	(84.5, 77.8)	(83.1, 76.2)	(80.7, 40.3)	1.1%
`person`	(45.1, 36.3)	(42.6, 34.1)	(32.8, 16.4)	(49.5, 40.0)	(47.6, 38.2)	(38.6, 19.3)	9.8%
`rider`	(35.1, 27.3)	(31.2, 24.0)	(18.6, 09.3)	(37.2, 29.2)	(33.9, 26.3)	(24.0, 12.0)	6.0%
`car`	(84.1, 76.9)	(83.4, 76.2)	(80.8, 40.4)	(84.0, 76.6)	(81.8, 74.0)	(81.2, 40.6)	$\sim$
`truck`	(24.7, 21.9)	(25.6, 22.7)	(21.8, 10.9)	(26.6, 23.3)	(28.1, 24.8)	(26.8, 13.4)	9.8%
`bus`	(46.8, 42.2)	(48.8, 43.8)	(36.3, 18.2)	(51.3, 46.5)	(51.5, 46.8)	(39.2, 19.6)	5.5%
`train`	(34.9, 30.7)	(36.3, 31.0)	(33.8, 16.9)	(35.8, 31.5)	(37.3, 32.2)	(34.7, 17.4)	2.8%
`motorcycle`	(19.7, 15.8)	(20.4, 16.1)	(07.0, 03.5)	(22.2, 17.7)	(21.1, 16.8)	(08.7, 04.4)	8.8%
`bicycle`	(41.4, 32.5)	(42.1, 32.9)	(32.9, 16.5)	(41.9, 32.6)	(42.0, 32.5)	(36.7, 18.4)	$\sim$

Table 5: Averaged class-specific Dice and IoU metrics of *Threshold/Argmax*, and the proposed *RankDice* based on various losses of PSPNet + resnet50 on **Fine-annotated CityScapes** *val* set. The class-specific improvement in terms of the Dice metric of the proposed RankDice framework is computed in "imps.", where $\sim$ indicates that the difference between *Threshold/Argmax* and *RankDice* is smaller than 1.0%.

6. Conclusions and future work

Summary. In this paper, we proposed a ranking-based framework for segmentation called *RankSEG* that comprises three steps: conditional probability estimation, ranking, and volume estimation. Specifically, we have focused on the Dice metric and developed *RankDice*, a version of *RankSEG* for optimal Dice-segmentation. We introduced a key concept "Dice-calibrated" and demonstrated that *RankDice* is able to recover the optimal segmentation rule, as opposed to the existing fixed-thresholding frameworks that are suboptimal with respect to the Dice metric. Computationally, we have developed efficient exact/approximate numerical methods, including GPU-enabled algorithms, to carry out *RankDice*. Moreover, we established general theoretical results, including excess risk bounds and a rate of convergence for *RankDice*, showing that *RankDice* is consistent when the conditional probability estimation is well-calibrated. Empirical experiments suggested that the proposed framework performs consistently well on a variety of segmentation benchmarks and state-of-the-art deep learning architectures. In parallel to *RankDice*, we also developed the framework *RankIoU* for the IoU metric. The theoretical results are similar, while the computation for the optimal IoU-segmentation could be more expensive in high-dimensional situation.

Limitation and future work. (i) For multiclass/multilabel segmentation, our results in this paper cover the overlapping (allowing) case; however, computing the optimal segmentation for the non-

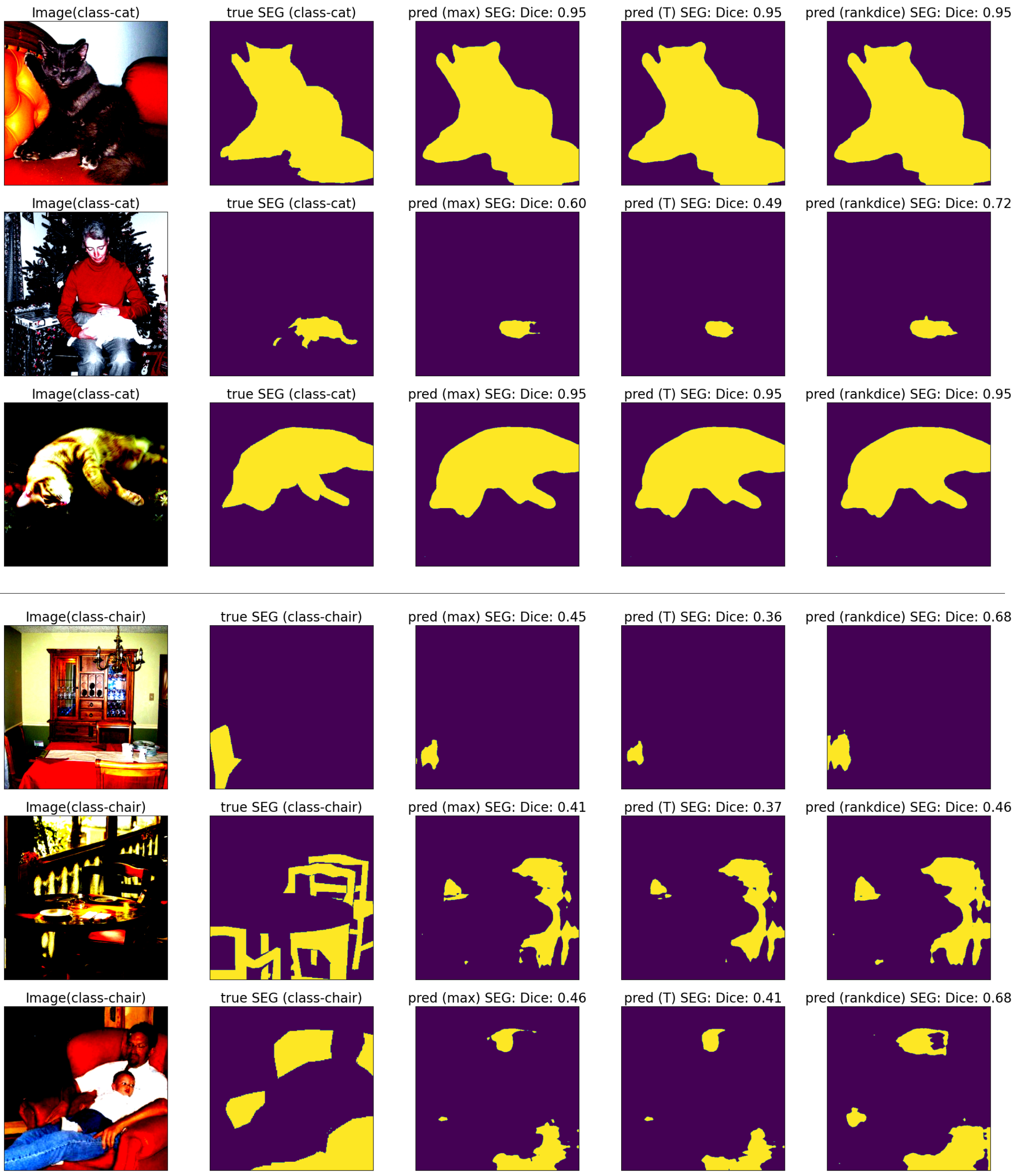

Figure 5: Comparison of segmentation results between the proposed method and existing methods for classes `cat` (upper panel) and `chair` (lower panel). Column 1 indicates original images, Column 2 indicates ground truths, and Columns 3-5 indicate the predicted segmentation produced by *argmax*, *thresholding*, and the proposed *RankDice*, respectively. The results are provided by a PSPNet trained with the cross-entropy loss.

Object class	Threshold/Argmax (Dice, IoU) (×.01) CE	Focal	BCE	RankDice (our) (Dice, IoU) (×.01) CE	Focal	BCE	**imps.** (best vs. best) (Dice)
`Aeroplane`	(71.2, 63.4)	(68.4, 59.2)	(72.9, 36.5)	(71.3, 63.4)	(72.7, 64.1)	(75.3, 37.6)	3.3%
`Bicycle`	(37.1, 25.9)	(19.6, 12.4)	(14.6, 7.30)	(38.7, 27.3)	(30.5, 20.6)	(23.1, 11.5)	4.3%
`Bird`	(76.0, 68.2)	(74.3, 65.2)	(74.2, 37.1)	(76.6, 68.7)	(75.8, 66.4)	(76.3, 38.1)	$\sim$
`Boat`	(51.1, 42.7)	(59.5, 49.1)	(55.5, 27.8)	(51.3, 42.9)	(61.9, 51.5)	(61.0, 30.5)	4.0%
`Bottle`	(42.8, 35.8)	(36.2, 30.0)	(39.1, 19.6)	(44.2, 36.8)	(37.6, 31.4)	(41.1, 20.6)	3.3%
`Bus`	(72.8, 68.3)	(72.3, 67.5)	(74.8, 37.4)	(74.1, 69.6)	(73.5, 68.8)	(75.9, 37.9)	1.5%
`Car`	(53.5, 47.5)	(51.1, 45.6)	(48.9, 24.4)	(55.0, 49.0)	(53.6, 47.9)	(51.7, 25.9)	2.7%
`Cat`	(75.0, 69.2)	(74.1, 67.9)	(73.1, 36.6)	(75.5, 69.7)	(75.4, 68.7)	(75.1, 37.6)	$\sim$
`Chair`	(17.5, 12.8)	(16.7, 11.6)	(10.2, 5.10)	(19.6, 14.4)	(22.2, 16.1)	(14.5, 7.30)	26.9%
`Cow`	(65.3, 58.6)	(60.1, 53.7)	(64.9, 32.4)	(66.5, 59.9)	(62.3, 56.0)	(68.4, 34.2)	4.8%
`Diningtable`	(32.9, 27.5)	(33.6, 27.4)	(31.7, 15.9)	(34.5, 29.2)	(38.6, 32.4)	(35.3, 17.6)	14.9%
`Dog`	(64.6, 57.9)	(71.0, 63.4)	(71.7, 35.9)	(65.5, 58.7)	(72.5, 64.9)	(74.4, 37.2)	3.8%
`Horse`	(63.9, 55.3)	(67.3, 58.3)	(67.0, 33.5)	(65.3, 56.6)	(69.5, 60.1)	(70.9, 35.4)	5.4%
`Motorbike`	(69.7, 60.6)	(65.5, 56.7)	(66.9, 33.5)	(71.6, 62.6)	(67.0, 57.9)	(70.1, 35.1)	2.7%
`Person`	(67.0, 57.7)	(65.0, 55.4)	(67.4, 33.7)	(67.4, 58.1)	(67.2, 57.6)	(69.7, 34.8)	3.4%
`Pottedplant`	(26.9, 20.2)	(22.4, 17.3)	(25.5, 12.8)	(29.1, 22.0)	(26.9, 20.7)	(28.6, 14.3)	8.2%
`Sheep`	(53.9, 47.4)	(62.8, 55.4)	(62.1, 31.1)	(54.3, 47.9)	(66.0, 58.6)	(66.9, 33.4)	6.5%
`Sofa`	(29.8, 25.0)	(29.8, 24.4)	(33.7, 16.8)	(32.0, 26.9)	(34.6, 29.0)	(38.9, 19.4)	14.5%
`Train`	(77.7, 71.0)	(75.8, 68.9)	(80.3, 40.1)	(77.9, 71.1)	(77.3, 70.4)	(82.1, 41.1)	2.2%
`Tvmonitor`	(48.4, 41.4)	(50.7, 41.6)	(53.7, 26.8)	(49.2, 42.0)	(54.1, 45.4)	(56.4, 28.2)	5.0%

Table 6: Class-specific Dice and IoU of *Threshold/Argmax*, and the proposed *RankDice* based on various losses of PSPNet + resnet50 on **PASCAL VOC 2012** *val* set. The class-specific improvement in terms of the Dice metric of the proposed *RankDice* framework is computed in "imps.", where $\sim$ indicates that the difference between *Threshold/Argmax* and the proposed *RankDice* is smaller than 1.0%.

Framework	Thold	(Dice, IoU) (×.01)
threshold-based	0.1	(49.10, 24.60)
	0.2	(53.00, 26.50)
	0.3	(53.60, 26.80)
	0.4	(52.90, 26.50)
	0.5	(51.40, 25.70)
	0.6	(49.60, 24.80)
	0.7	(47.00, 23.50)
	0.8	(43.40, 21.70)
	0.9	(37.40, 18.70)
RankDice(our)	—	**(55.10, 27.60)**

Framework	Thold	(Dice, IoU) (×.01)
threshold-based	0.1	(56.80, 28.40)
	0.2	(63.90, 32.00)
	0.3	(65.70, 32.80)
	0.4	(65.60, 32.80)
	0.5	(64.20, 32.10)
	0.6	(62.30, 32.00)
	0.7	(59.30, 29.60)
	0.8	(54.20, 27.10)
	0.9	(43.40, 21.70)
RankDice(our)	—	**(67.10, 33.50)**

Table 7: The averaged Dice and IoU metrics and their standard errors (in parentheses) of the proposed *RankDice* framework and the fixed-*thresholding* (with different thresholds) framework in **Fine-annotated CityScapes** (left) and **PASCAL VOC 2012** (right) datasets. The performance is reported based on PSPNet + resnet50 with the BCE loss.

overlapping case is NP-hard. Thus, it would be interesting to develop a scalable approximating algorithm to utilize the proposed framework in the non-overlapping setting. (ii) The conditional independence, $Y_i \perp Y_j | \mathbf{X}$ for any $i \neq j$, is crucial for Theorem 1 and subsequent theorems in Section 4. In some segmentation applications, it is of interest to extend the proposed frameworks and theorems with locally dependent outcomes. (iii) When given the testing features, the proposed method can be extended to maximize the F1-score in classification tasks.

Dataset	Temp	Threshold (Dice, IoU) ($\times$.01) CE	Threshold (Dice, IoU) ($\times$.01) BCE	RankDice (our) (Dice, IoU) ($\times$.01) CE	RankDice (our) (Dice, IoU) ($\times$.01) BCE
CityScapes	1.0	(**57.50**, **49.60**)	(51.40, 25.70)	(59.30, 51.00)	(**55.10**, **27.60**)
	1.2	(57.40, 49.60)	/	(**59.40**, **51.20**)	(54.70, 26.30)
	1.5	(56.90, 49.20)	/	(59.40, 51.20)	(50.60, 25.30)
	1.7	(56.20, 48.50)	/	(59.10, 51.00)	(47.20, 23.60)
	2.0	(54.40, 46.90)	/	(58.00, 50.10)	(47.20, 23.60)
	2.2	(52.80, 45.40)	/	(57.10, 49.30)	(44.70, 22.40)
	2.5	(49.80, 42.50)	/	(55.40, 48.00)	(41.50, 20.70)
VOC 2012	1.0	(64.60, 57.10)	(64.20, 32.10)	(65.40, 57.80)	(67.10, 33.50)
	1.2	(64.50, 57.50)	/	(66.10, 58.30)	(68.80, 34.40)
	1.5	(**65.30**, **57.70**)	/	(66.90, 59.10)	(**69.30**, **34.60**)
	1.7	(65.30, 57.70)	/	(67.60, 59.80)	(69.00, 34.50)
	2.0	(64.80, 57.10)	/	(68.30, 60.50)	(68.00, 34.00)
	2.2	(63.80, 56.00)	/	(**68.40**, **60.60**)	(67.00, 33.50)
	2.5	(61.30, 53.40)	/	(67.90, 60.20)	(65.10, 32.50)

Table 8: The averaged Dice and IoU metrics and their standard errors (in parentheses) of the proposed *RankDice* framework and the fixed-*thresholding* framework based on temperature-scaling calibration methods with different temperature tuning parameters in **Fine-annotated CityScapes** and **PASCAL VOC 2012** datasets. '/' indicates that the performance of the thresholding-based framework over different temperatures are all the same under BCE loss. The performance is reported based on PSPNet + resnet50.

Acknowledgments

We would like to acknowledge support for this project from the HK RGC-ECS 24302422 and the CUHK direct fund. Ben Dai developed the main framework (*RankDice*), theory, algorithms and experiments. Chunlin Li mainly extended *RankDice* to *RankIoU* (Section 4.2), and partly contributed to the proof of Lemma 3 and Theorem 11. We would like to thank the referees and the Action Editor for constructive feedback which greatly improved this work.

Appendix A. Empirical evaluation of the Dice metric

Recall the definition of the Dice and IoU metrics in (1), their empirical evaluation based on a validation/testing dataset $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)_{i=1,\cdots,m}$, can be written as:

$$\begin{aligned}
\widehat{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}) &= \frac{1}{m}\sum_{i=1}^m \frac{2\tilde{\mathbf{y}}_i^\intercal \boldsymbol{\delta}(\tilde{\mathbf{x}}_i) + \gamma}{\|\tilde{\mathbf{y}}_i\|_1 + \|\boldsymbol{\delta}(\tilde{\mathbf{x}}_i)\|_1 + \gamma} = \frac{1}{m}\sum_{i=1}^m \frac{2\mathrm{TP}_i + \gamma}{2\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i + \gamma}, \\
\widehat{\mathrm{IoU}}_\gamma(\boldsymbol{\delta}) &= \frac{1}{m}\sum_{i=1}^m \Big(\frac{\tilde{\mathbf{y}}_i^\intercal \boldsymbol{\delta}(\tilde{\mathbf{x}}_i) + \gamma}{\|\tilde{\mathbf{y}}_i\|_1 + \|\boldsymbol{\delta}(\tilde{\mathbf{x}}_i)\|_1 - \tilde{\mathbf{y}}_i^\intercal \boldsymbol{\delta}(\tilde{\mathbf{x}}_i) + \gamma}\Big) = \frac{1}{m}\sum_{i=1}^m \frac{\mathrm{TP}_i + \gamma}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i + \gamma},
\end{aligned} \tag{22}$$

where TP_i, FP_i and FN_i are defined at the instance level. In general, the empirical Dice and IoU metrics are not equal to the evaluation criteria used in some literature:

$$\begin{aligned}
\widehat{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}) \neq \overline{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}) &:= \frac{\frac{1}{m}\sum_{i=1}^m 2\tilde{\mathbf{y}}_i^\intercal \boldsymbol{\delta}(\tilde{\mathbf{x}}_i) + \gamma}{\frac{1}{m}\sum_{i=1}^m \|\tilde{\mathbf{y}}_i\|_1 + \frac{1}{m}\sum_{i=1}^m \|\boldsymbol{\delta}(\tilde{\mathbf{x}}_i)\|_1 + \gamma} \xrightarrow{\mathbb{P}} \frac{\mathbb{E}\big(2\mathbf{Y}^\intercal\boldsymbol{\delta}(\mathbf{X})\big) + \gamma}{\mathbb{E}\big(\|\mathbf{Y}\|_1\big) + \mathbb{E}\big(\|\boldsymbol{\delta}(\mathbf{X})\|_1\big) + \gamma}, \\
\widehat{\mathrm{IoU}}_\gamma(\boldsymbol{\delta}) \neq \overline{\mathrm{IoU}}_\gamma(\boldsymbol{\delta}) &:= \frac{\frac{1}{m}\sum_{i=1}^m \tilde{\mathbf{y}}_i^\intercal \boldsymbol{\delta}(\tilde{\mathbf{x}}_i) + \gamma}{\frac{1}{m}\sum_{i=1}^m \|\tilde{\mathbf{y}}_i\|_1 + \frac{1}{m}\sum_{i=1}^m \|\boldsymbol{\delta}(\tilde{\mathbf{x}}_i)\|_1 - \frac{1}{m}\sum_{i=1}^m \tilde{\mathbf{y}}_i^\intercal \boldsymbol{\delta}(\tilde{\mathbf{x}}_i) + \gamma} \\
&\xrightarrow{\mathbb{P}} \frac{\mathbb{E}\big(2\mathbf{Y}^\intercal\boldsymbol{\delta}(\mathbf{X})\big) + \gamma}{\mathbb{E}\big(\|\mathbf{Y}\|_1\big) + \mathbb{E}\big(\|\boldsymbol{\delta}(\mathbf{X})\|_1\big) - \mathbb{E}\big(\mathbf{Y}^\intercal\boldsymbol{\delta}(\mathbf{X})\big) + \gamma}.
\end{aligned} \tag{23}$$

Here $\xrightarrow{\mathbb{P}}$ denotes convergence in probability following from the law of large numbers and Slutsky's theorem. Clearly, both empirical and population evaluations in (23) do not match with the empirical verisons in (22) and the population Dice in (1). Although the empirical evaluation in (23) is widely used, it inherently discounts the effects of instances with small segmented features/pixels, leading to bias in the empirical evaluation. The issues of $\overline{\mathrm{Dice}}_\gamma(\boldsymbol{\delta})$ and $\overline{\mathrm{IoU}}_\gamma(\boldsymbol{\delta})$ are also indicated in some recent literature, including Cordts et al. (2016) and Berman et al. (2018).

Therefore, it is highly recommended using the empirical Dice in (22) in implementation, and our numerical results in Section 5 are reported based on (22).

Appendix B. Auxiliary definitions

B.1 Population RankSEG

In this section, we present the definition of population *RankSEG*, including the proposed frameworks *RankDice* and *RankIoU*. In other words, we work with population of $(\mathbf{X}, \mathbf{Y}) \in \mathbb{R}^d \times \{0,1\}^d$. Denote $\mathcal{Q}$ as the class of all measurable functions $\mathbf{q} : \mathbf{x} \in \mathbb{R}^d \to \mathbf{q}(\mathbf{x}) = (q_1(\mathbf{x}), \cdots, q_d(\mathbf{x}))^\intercal \in [0,1]^d$.

Step 1 (Conditional probability estimation): Estimate the conditional probability based on a strictly proper loss $l(\cdot,\cdot)$:

$$\widehat{\mathbf{q}} = \underset{\mathbf{q}\in\mathcal{Q}}{\mathrm{argmin}}\, \mathbb{E}\big(l(\mathbf{Y}, \mathbf{q}(\mathbf{X}))\big). \tag{24}$$

Step 2 (Ranking): Given a new instance $\mathbf{x}$, sort its estimated conditional probabilities decreasingly, and denote the corresponding indices as $j_1, \cdots, j_d$, that is, $\widehat{q}_{j_1}(\mathbf{x}) \geq \widehat{q}_{j_2}(\mathbf{x}) \geq \cdots \geq \widehat{q}_{j_d}(\mathbf{x})$.

Step 3 (Volume estimation): From (4), we estimate the volume $\widehat{\tau}(\mathbf{x})$ by replacing the true conditional probability $\mathbf{p}(\mathbf{x})$ by the estimated one $\widehat{\mathbf{q}}(\mathbf{x})$:

$$\text{(RankDice)} \quad \widehat{\tau}(\mathbf{x}) = \operatorname*{argmax}_{\tau \in \{0,\cdots,d\}} \sum_{s=1}^{\tau} \sum_{l=0}^{d-1} \frac{2}{\tau + l + \gamma + 1} \widehat{q}_{j_s}(\mathbf{x}) \mathbb{P}\big(\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x}) = l\big) + \sum_{l=0}^{d} \frac{\gamma}{\tau + l + \gamma} \mathbb{P}\big(\widehat{\Gamma}(\mathbf{x}) = l\big),$$

$$\text{(RankIoU)} \quad \widehat{\tau}(\mathbf{x}) = \operatorname*{argmax}_{\tau \in \{0,1,\cdots,d\}} \Big(\sum_{j \in J_\tau(\mathbf{x})} \widehat{q}_j(\mathbf{x}) + \gamma \Big) \sum_{l=0}^{d-\tau} \frac{1}{\tau + l + \gamma} \mathbb{P}\big(\widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = l\big),$$

where $\widehat{\Gamma}(\mathbf{x}) = \sum_{j=1}^{d} \widehat{B}_j(\mathbf{x})$, $\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x}) = \sum_{j \neq j_s} \widehat{B}_j(\mathbf{x})$, and $\widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = \sum_{j \notin J_\tau(\mathbf{x})} \widehat{B}_j(\mathbf{x})$ denote Poisson-binomial random variables, and $\widehat{B}_j(\mathbf{x})$ is a Bernoulli random variable with the success probability $\widehat{q}_j(\mathbf{x})$; for $j = 1, \cdots, d$.

B.2 Poisson-binomial distribution

The Poisson binomial distribution is the discrete probability distribution of a sum of independent non-identical Bernoulli trials. Specifically, suppose $B_1, \cdots, B_d$ are independent Bernoulli random variables, with probabilities of success $\mathbf{p} = (p_1, \cdots, p_d)^\intercal$, then $\Gamma = \sum_{j=1}^{d} B_j$ is a Poisson-Binomial random variable with parameter $\mathbf{p}$, denoted as $\Gamma \sim \text{PB}(\mathbf{p})$, and its probability mass function is:

$$\mathbb{P}\big(\Gamma = l\big) = \sum_{\mathbf{b}: \|\mathbf{b}\|_1 = l} \prod_{j=1}^{d} \big(b_j p_j + (1 - b_j)(1 - p_j)\big),$$

where $\mathbf{b} = (b_1, \cdots, b_d)^\intercal \in \{0,1\}^d$. Moreover, the mean, variance, and skewness for $\Gamma \sim \text{PB}(\mathbf{p})$ are listed as follows.

$$\mu := \mathbb{E}(\Gamma) = \sum_{j=1}^{d} p_j, \ \sigma^2 := \text{Var}(\Gamma) = \sum_{j=1}^{d} p_j(1 - p_j), \ \eta := \text{Skew}(\Gamma) = \frac{1}{\sigma^3} \sum_{j=1}^{d} p_j(1 - p_j)(1 - 2p_j).$$

B.3 Conditional independence in segmentation

In this section, we adopt a probabilistic perspective on the likelihood of the segmentation task, to suggest that the conditional independence ($Y_j \perp Y_{j'} \mid \mathbf{X}$ for $j \neq j'$) is implicitly assumed to ensure the validity of the cross-entropy (CE) loss, and widely accepted due to the high dimensional nature of segmentation data.

Suppose $(\mathbf{X}_i, \mathbf{Y}_i) \overset{\text{iid}}{\sim} \mathbb{P}_{\mathbf{X},\mathbf{Y}}$, the negative conditional log-likelihood function of $\mathbf{q}$ for the probabilistic model is:

$$\begin{aligned} \mathcal{L}_n(\mathbf{q}) &:= -\log\Big(\prod_{i=1}^{n} \mathbb{P}_{\mathbf{q}}(\mathbf{Y} = \mathbf{y}_i \mid \mathbf{X} = \mathbf{x}_i)\Big) = -\log\Big(\prod_{i=1}^{n}\prod_{j=1}^{d} q_j(\mathbf{x}_i)^{y_{ij}} (1 - q_j(\mathbf{x}_i))^{1 - y_{ij}}\Big) \\ &= -\sum_{i=1}^{n}\sum_{j=1}^{d} \big(y_{ij} \log(q_j(\mathbf{x}_i)) + (1 - y_{ij}) \log(1 - q_j(\mathbf{x}_i))\big) = \sum_{i=1}^{n} l_{\text{CE}}(\mathbf{y}_i, \mathbf{q}(\mathbf{x}_i)), \end{aligned}$$

where the second equality follows from the conditional independence assumption $Y_j \perp Y_{j'} \mid \mathbf{X}$ for $j \neq j'$, which connects the CE loss to the negative conditional log-likelihood function. In this sense, the conditional independence is naturally (and implicitly) assumed by CE to presuppose the probabilistic interpretation of its estimator.

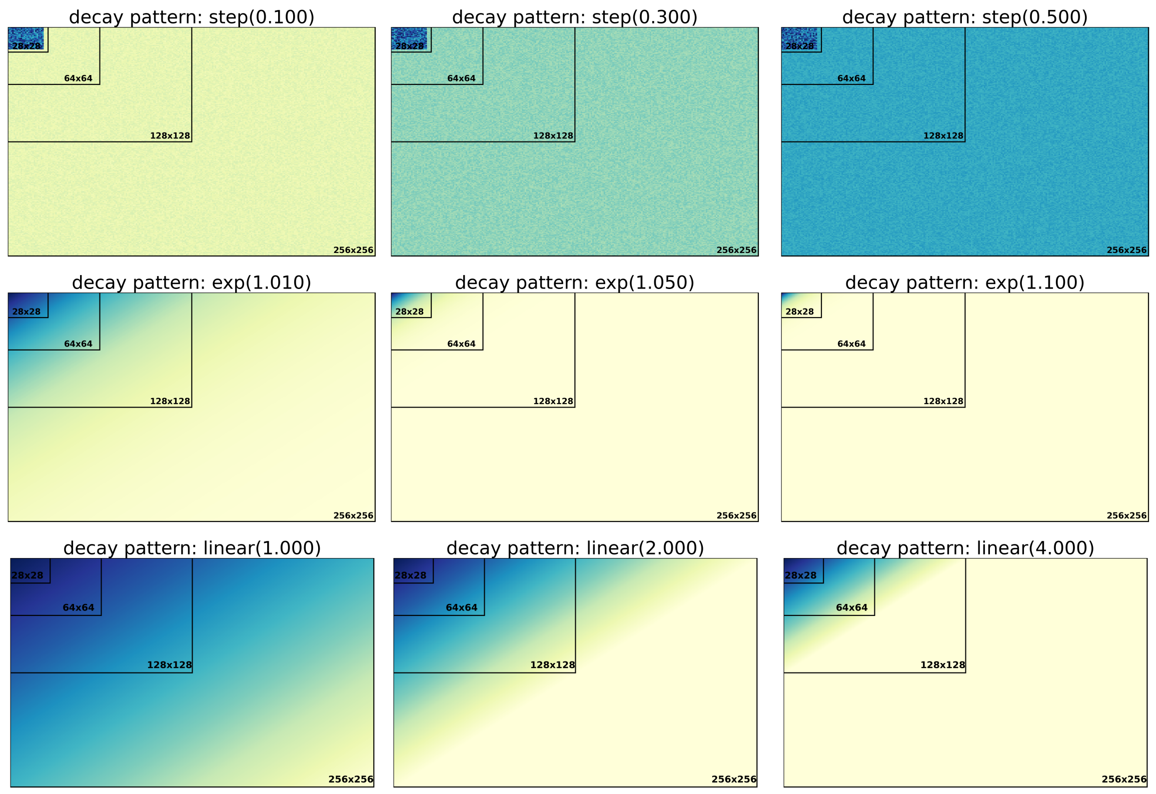

Figure 6: Simulation setting in Section 5.1 with different decay patterns and dimensions/shapes (28x28 - 256x256). **Upper panel.** Heatmaps for the simulated probabilities with step decay ($\beta = 0.1, 0.3, 0.5$). **Middle panel.** Heatmaps for the simulated probabilities with exponential decay (base=1.01, 1.05, 1.10). **Lower panel.** Heatmaps for the simulated probabilities with linear decay (slope=1, 2, 4). The performance for the proposed *RankDice* and *thresholding*-based frameworks is summarized in Table 9.

Moreover, the conditional independence is widely accepted due to the high dimensional nature of segmentation data. For example, given a 512x512 image, it is infeasible to consider 512^4 pairs of label-dependence, which can even be adaptive with respect to $\mathbf{x}$.

Appendix C. Simulation results and Implementation details

C.1 Simulation setting and results

This subsection includes the simulation setting demonstration (Figure 6) and the numerical results (Tables 9 and 10, and Figure 7).

Decay	Shape	Threshold (at 0.5)	RankDice (our)
step(0.1)	28x28	0.049(.000)	0.274(.001)
	64x64	0.083(.000)	0.279(.000)
	128x128	0.081(.000)	0.278(.000)
	256x256	0.089(.000)	0.279(.000)
step(0.3)	28x28	0.022(.001)	0.499(.001)
	64x64	0.038(.000)	0.517(.001)
	128x128	0.036(.000)	0.518(.000)
	256x256	0.040(.000)	0.518(.000)
step(0.5)	28x28	0.708(.000)	0.708(.000)
	64x64	0.707(.000)	0.707(.000)
	128x128	0.708(.000)	0.708(.000)
	256x256	0.708(.000)	0.708(.000)
exp(1.01)	28x28	0.870(.000)	0.870(.000)
	64x64	0.669(.000)	0.714(.000)
	128x128	0.410(.000)	0.551(.000)
	256x256	0.286(.000)	0.450(.000)
exp(1.05)	28x28	0.427(.001)	0.551(.001)
	64x64	0.296(.001)	0.446(.001)
	128x128	0.276(.001)	0.427(.001)
	256x256	0.274(.001)	0.427(.001)
exp(1.10)	28x28	0.332(.002)	0.467(.002)
	64x64	0.301(.001)	0.439(.002)
	128x128	0.300(.002)	0.438(.002)
	256x256	0.298 (.002)	0.436(.002)

Decay	Shape	Threshold (at 0.5)	RankDice (our)
linear(1.00)	28x28	0.679(.001)	0.717(.001)
	64x64	0.672(.000)	0.711(.000)
	128x128	0.669(.000)	0.709(.000)
	256x256	0.668(.000)	0.707(.000)
linear(2.00)	28x28	0.578(.001)	0.647(.001)
	64x64	0.575(.001)	0.642(.001)
	128x128	0.573(.000)	0.638(.000)
	256x256	0.573(.000)	0.637(.000)
linear(4.00)	28x28	0.588(.003)	0.663(.002)
	64x64	0.580(.001)	0.646(.001)
	128x128	0.575(.001)	0.642(.001)
	256x256	0.574(.000)	0.639(.000)

Table 9: The averaged Dice metrics and its standard errors (in parentheses) of the proposed *RankDice* framework and the *thresholding*-based (or *argmax*-based) framework in Example 1 (see Fig 6) in Section 5.1.

Framework	Threshold	Dice
threshold-based	0.1	0.481(.005)
	0.2	0.560(.005)
	0.3	0.560(.005)
	0.4	0.560(.005)
	0.5	0.560(.005)
	0.6	0.528(.005)
	0.7	0.471(.005)
	0.8	0.377(.005)
	0.9	0.230(.004)
RankDice(our)	—	**0.601(0.005)**

Table 10: The averaged Dice metrics and its standard errors (in parentheses) of the proposed *RankDice* framework and the *thresholding*-based (with different thresholds) framework in Example 2 in Section 5.1.

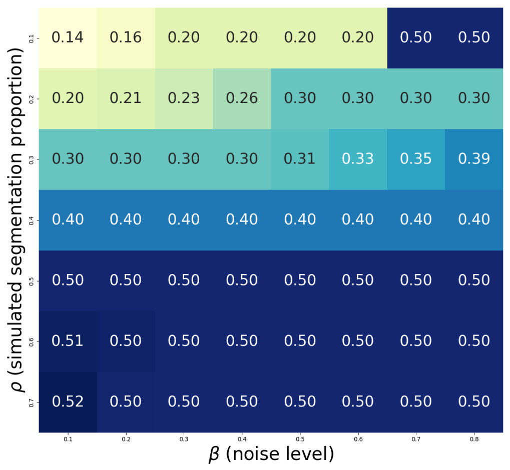

Figure 7: The optimal thresholds for different images with various generating parameters (β, ρ) in Example 2 in Section 5.1.

C.2 Implementation details

The experiment protocol of our numerical sections basically follows a well-developed Github repository PYTORCH-SEGMENTATION (Ouali, 2022). The major difference lies in the empirical evaluation of the Dice and IoU metrics. In our experiments, we report the unbiased evaluations $\widehat{\text{mDice}}_\gamma(\cdot)$

and $\widehat{\text{mIoU}}_\gamma(\cdot)$, yet the biased evaluations $\overline{\text{mDice}}_\gamma(\cdot)$ and $\overline{\text{mIoU}}_\gamma(\cdot)$ are usually used for existing literature, see more discussion and definitions in Appendix A.

To justify the effectiveness of our experiment protocol, we also report $\overline{\text{mDice}}_\gamma(\cdot)$ and $\overline{\text{mIoU}}_\gamma(\cdot)$ under our setting based on the *argmax*-based framework and compare the performance with the existing benchmarks. Specifically, in our setting, the performance is, **DeepLab**: (i) CityScapes: $\overline{\text{mIoU}}$ 63.20%; $\overline{\text{mDice}}$ 76.10%; (ii) VOC: $\overline{\text{mIoU}}$ 74.40%; $\overline{\text{mDice}}$ 84.30%; **PSPNet**: (i) CityScapes: $\overline{\text{mIoU}}$ 65.20%; $\overline{\text{mDice}}$ 77.60%; (ii) VOC *test*: $\overline{\text{mIoU}}$ 79% (which is provided by Ouali (2022) with the same configuration expect `batch_size=16`); **FCN8**: VOC: $\overline{\text{mIoU}}$ 55.60%; $\overline{\text{mDice}}$ 70.40%.

The experiment protocol, including `learning_rate`, `crop_size`, `backbone`, and `batch_size`, on the existing networks are summarized as follows.

DeepLab (Chen et al., 2018). The experiment on the Fine-annotated CityScapes dataset is set as follows: `backbone` is "Xception-65". The final $\overline{\text{mIoU}}$ is 79.14%; The experiment on the PASCAL VOC 2012 dataset is set as follows: `learning_rate` is 0.007 with `poly` schedule; `crop_size` is 513x513, `backbone` is "resnet101", `batch_size` is 16. The final $\overline{\text{mIoU}}$ is 78.21%;
PSPNet (Zhao et al., 2017). The experiment on the Fine-annotated CityScapes dataset is set as follows: `learning_rate` is 0.01. The final $\overline{\text{mIoU}}$ is 78.4%; The experiment on the PASCAL VOC 2012 dataset is set as follows: `learning_rate` is 0.01, `batch_size` is 16. The final $\overline{\text{mIoU}}$ based on the VOC *test* datset is 82.6%;
FCN8 (Long et al., 2015). The experiment on the PASCAL VOC 2012 dataset is set as follows: `learning_rate` is 0.0001, `batch_size` is 20, `backbone` is "VGG16". The final $\overline{\text{mIoU}}$ is 62.2%;

Note that the suboptimal performance of our experiment may be caused by a small batch/crop size, different specified backbone models, and other fitting hyperparameters. The current experiment can be further improved by carefully tuning the hyperparameters, yet it provides a fair numerical comparison of all frameworks (threshold-based, argmax-based, and the proposed *RankDice*).

Appendix D. Technical proofs

D.1 Proof of Theorem 1

Proof It suffices to consider the point-wise maximization:

$$\boldsymbol{\delta}^*(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{v}\in\{0,1\}^d} \text{Dice}_\gamma(\mathbf{v}|\mathbf{x}), \quad \text{Dice}_\gamma(\mathbf{v}|\mathbf{x}) = \mathbb{E}\Big(\frac{2\mathbf{Y}^\intercal\mathbf{v}+\gamma}{\|\mathbf{Y}\|_1+\|\mathbf{v}\|_1+\gamma}\Big|\mathbf{X}=\mathbf{x}\Big).$$

Let $\mathbf{y}_{\text{-}j} = (y_1,\cdots,y_{j-1},y_{j+1},\cdots,y_d)^\intercal$, $I(\mathbf{v}) = I(\boldsymbol{\delta}(\mathbf{x})) = \{j : v_j = 1\}$ be the index set of segmented features by $\boldsymbol{\delta}(\mathbf{x})$, and $\|\mathbf{v}\|_1 = \tau$, we have

$$\text{Dice}_\gamma(\mathbf{v}|\mathbf{x}) = \mathbb{E}\Big(\frac{2\mathbf{Y}^\intercal\mathbf{v}}{\|\mathbf{Y}\|_1+\tau+\gamma}\Big|\mathbf{X}=\mathbf{x}\Big) + \mathbb{E}\Big(\frac{\gamma}{\|\mathbf{Y}\|_1+\tau+\gamma}\Big|\mathbf{X}=\mathbf{x}\Big).$$

Note that the second term is only related to τ, and the first term can be rewritten as:

$$\begin{aligned}\mathbb{E}\Big(\frac{2\mathbf{Y}^\intercal\mathbf{v}}{\|\mathbf{Y}\|_1+\tau+\gamma}\Big|\mathbf{X}=\mathbf{x}\Big) &= \sum_{\mathbf{y}\in\{0,1\}^d}\frac{2\mathbf{y}^\intercal\mathbf{v}\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x})}{\tau+\|\mathbf{y}\|_1+\gamma} = \sum_{\mathbf{y}\in\{0,1\}^d}\sum_{j=1}^d\frac{2y_jv_j\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x})}{\tau+\|\mathbf{y}\|_1+\gamma} \\ &= \sum_{j\in I(\mathbf{v})}\sum_{\mathbf{y}\in\{0,1\}^d}\frac{2y_j\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x})}{\tau+\|\mathbf{y}\|_1+\gamma} = \sum_{j\in I(\mathbf{v})}\sum_{\substack{\mathbf{y}_{\text{-}j}\in\{0,1\}^{d-1}\\ y_j=1}}\frac{2\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x})}{\tau+\|\mathbf{y}\|_1+\gamma} = \sum_{j\in I(\mathbf{v})} s_j(\tau).\end{aligned} \tag{25}$$

As indicated in (25), when τ is given, $\text{Dice}_\gamma(\mathbf{v}|\mathbf{x})$ is an additive function with respect to $j \in I(\mathbf{v})$. Therefore, maximizing $\text{Dice}_\gamma(\mathbf{v}|\mathbf{x})$ suffices to find the indices of top τ largest $s_j(\tau)$. Toward this end, we consider the differenced score function:

$$\begin{aligned}
D_{jj'}(\tau) = s_j(\tau) - s_{j'}(\tau) &= \sum_{\substack{\mathbf{y}_{\text{-}j}\in\{0,1\}^{d-1}\\ y_j=1}} \frac{2\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x})}{\tau+\|\mathbf{y}\|_1+\gamma} - \sum_{\substack{\mathbf{y}_{\text{-}j'}\in\{0,1\}^{d-1}\\ y_{j'}=1}} \frac{2\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x})}{\tau+\|\mathbf{y}\|_1+\gamma} \\
&= \sum_{\substack{\mathbf{y}_{\text{-}jj'}\in\{0,1\}^{d-2}\\ y_j=1;y_{j'}=0}} \frac{2\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x})}{\tau+\|\mathbf{y}\|_1+\gamma} - \sum_{\substack{\mathbf{y}_{\text{-}jj'}\in\{0,1\}^{d-2}\\ y_j=0,y_{j'}=1}} \frac{2\mathbb{P}(\mathbf{Y}=\mathbf{y}|\mathbf{x})}{\tau+\|\mathbf{y}\|+\gamma} \\
&= \sum_{\substack{\mathbf{y}_{\text{-}jj'}\in\{0,1\}^{d-2}\\ y_j=1;y_{j'}=0}} \frac{2\prod_{i\neq\{j,j'\}}\mathbb{P}(Y_i=y_i|\mathbf{x})\mathbb{P}(Y_j=1|\mathbf{x})\mathbb{P}(Y_{j'}=0|\mathbf{x})}{\tau+1+\|\mathbf{y}_{\text{-}jj'}\|_1+\gamma} \\
&\qquad - \sum_{\substack{\mathbf{y}_{\text{-}jj'}\in\{0,1\}^{d-2}\\ y_j=0;y_{j'}=1}} \frac{2\prod_{i\neq\{j,j'\}}\mathbb{P}(Y_i=y_i|\mathbf{x})\mathbb{P}(Y_j=0|\mathbf{x})\mathbb{P}(Y_{j'}=1|\mathbf{x})}{\tau+1+\|\mathbf{y}_{\text{-}jj'}\|_1+\gamma} \\
&= \big(\mathbb{P}(Y_j=1|\mathbf{x}) - \mathbb{P}(Y_{j'}=1|\mathbf{x})\big) \sum_{\mathbf{y}_{\text{-}jj'}\in\{0,1\}^{d-2}} \frac{2\prod_{i\neq j,j'}\mathbb{P}(Y_i=y_i|\mathbf{x})}{\tau+1+\|\mathbf{y}_{\text{-}jj'}\|_1+\gamma},
\end{aligned} \tag{26}$$

where $\mathbf{y}_{\text{-}jj'}$ is $\mathbf{y}$ removing y_j and $y_{j'}$, the second last equality follows from the conditional independence of Y_j and $Y_{j'}$ given $\mathbf{X}$ for any pair j and j'. According to (26), we have

$$D_{jj'}(\tau) \geq 0 \quad \Longleftrightarrow \quad \mathbb{P}(Y_j=1|\mathbf{x}) - \mathbb{P}(Y_{j'}=1|\mathbf{x}) \geq 0,$$

thus sorting $s_j(\tau)$ is equivalent to sorting $\mathbb{P}(Y_j=1|\mathbf{x})$ for any given τ. Let $J_\tau = \big\{j : \sum_{j'=1}^d \mathbf{1}\big(\mathbb{P}(Y_{j'}=1|\mathbf{x}) \geq \mathbb{P}(Y_j=1|\mathbf{x})\big) \leq \tau\big\}$ be the index set of the τ-largest conditional probabilities, it suffices to solve

$$\begin{aligned}
\tau^* &= \operatorname*{argmax}_{\tau=0,\cdots,d} \sum_{j\in J_\tau} \mathbb{E}\big(\frac{2Y_j}{\|\mathbf{Y}\|_1+\tau+\gamma}\big) + \mathbb{E}\big(\frac{\gamma}{\|\mathbf{Y}\|_1+\tau+\gamma}\big) \\
&= \operatorname*{argmax}_{\tau=0,\cdots,d} \sum_{j\in J_\tau}\sum_{l=0}^{d-1} \frac{2p_j(\mathbf{x})}{\tau+l+\gamma+1}\mathbb{P}\big(\|\mathbf{Y}_{\text{-}j}\|_1 = l|\mathbf{X}=\mathbf{x}\big) + \sum_{l=0}^{d} \frac{\gamma}{\tau+l+\gamma}\mathbb{P}\big(\|\mathbf{Y}\|_1=l|\mathbf{X}=\mathbf{x}\big),
\end{aligned}$$

where $\|\mathbf{Y}_{\text{-}j}\|_1 = \sum_{j'\neq j} Y_{j'}$ is a Poisson-binomial random variable with success probabilities $\mathbf{p}_{\text{-}j}(\mathbf{x})$, since Y_j $(j=1,\cdots,d)$ is an independent Bernoulli random variable given $\mathbf{X}=\mathbf{x}$. The desirable result then follows. ■

D.2 Proof of Lemma 3

Proof To proceed, we denote $\xi_l(\mathbf{x}) = \mathbb{P}(\widehat{\Gamma}(\mathbf{x})=l)$, and $\xi_{jl}(\mathbf{x}) = \mathbb{P}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x})=l)$, and $\bar{\pi}_\tau = \bar{\omega}_\tau(\mathbf{x}) + \bar{\nu}_\tau(\mathbf{x})$. For simplicity in presentation, we assume that $\widehat{q}_1(\mathbf{x}) \geq \cdots \geq \widehat{q}_d(\mathbf{x})$. Then, for any $\tau' > \tau$, since $\bar{v}_\tau \geq \bar{v}_{\tau'}$, and

$$\begin{aligned}\frac{\bar{\pi}_\tau(\mathbf{x}) - \bar{\pi}_{\tau'}(\mathbf{x})}{\tau' - \tau} &\geq \sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) \sum_{l=0}^{d-1} \frac{\xi_{jl}(\mathbf{x})}{(\tau+l+1+\gamma)(\tau'+l+1+\gamma)} - \frac{1}{\tau'-\tau} \sum_{j=\tau+1}^{\tau'} \widehat{q}_j(\mathbf{x}) \sum_{l=0}^{d-1} \frac{\xi_{jl}(\mathbf{x})}{\tau'+l+\gamma+1} \\ &\geq \sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) \sum_{l=0}^{d-1} \frac{\xi_{\tau l}(\mathbf{x})}{(\tau+l+\gamma+1)(\tau'+l+\gamma+1)} - \widehat{q}_{\tau+1}(\mathbf{x}) \sum_{l=0}^{d-1} \frac{\xi_{\tau l}(\mathbf{x})}{\tau'+l+\gamma+1} \\ &\geq \widehat{q}_{\tau+1}(\mathbf{x}) \sum_{l=0}^{d-1} \frac{(\tau+\gamma+d)\xi_{\tau l}(\mathbf{x})}{(\tau+l+\gamma+1)(\tau'+l+\gamma+1)} - \widehat{q}_{\tau+1}(\mathbf{x}) \sum_{l=0}^{d-1} \frac{\xi_{\tau l}(\mathbf{x})}{\tau'+l+\gamma+1} \geq 0,\end{aligned} \tag{27}$$

where the first inequality follows from Lemma 17 with $\zeta_l = (\tau+l+\gamma+1)(\tau'+l+\gamma+1)$ and $\zeta_l = \tau'+l+\gamma+1$, and $\widehat{q}_1(\mathbf{x}) \geq \cdots \geq \widehat{q}_\tau(\mathbf{x}) \geq \cdots \geq \widehat{q}_{\tau'}(\mathbf{x})$, and the third inequality follows from the condition that $\sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x})/\widehat{q}_{\tau+1}(\mathbf{x}) \geq \tau+\gamma+d$. The desirable result then follows. The upper bound provided by Lemma 3 is illustrated in Figure 8 based on a random example of Example 1. ■

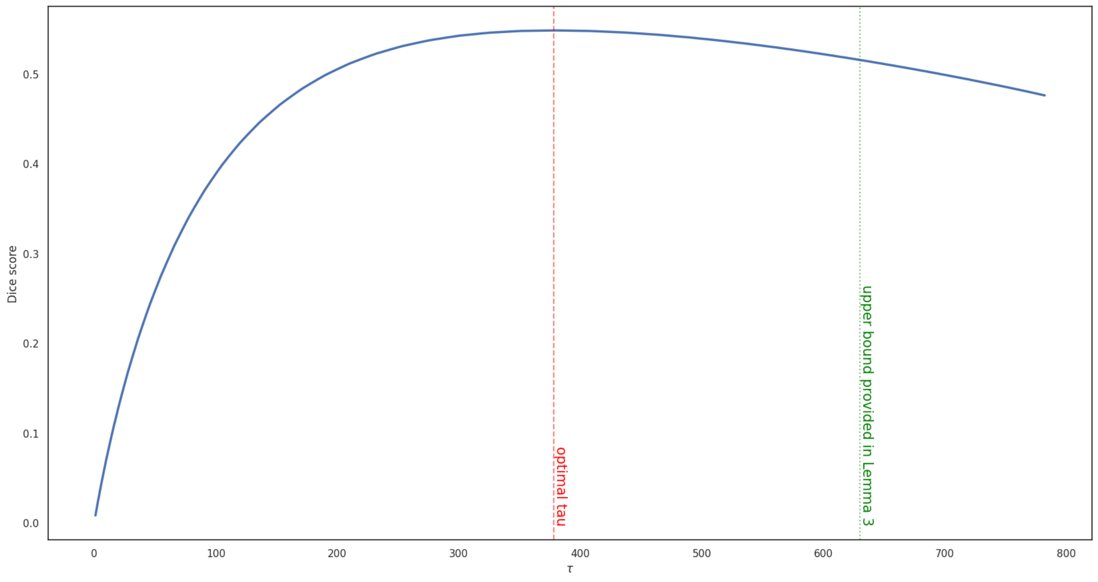

Figure 8: Dice score vs. τ based on a random example of Example 1. Note that Lemma 3 indicates the optimal τ (red line) is always obtained before the upper bound (green line). Thus, the searching region of τ can be shrunk.

D.3 Proof of Lemma 4

Proof Without loss of generality, assume $\mathcal{L}(\varepsilon) \subset \{0, \cdots, d-1\}$. Denote $l_L = \lfloor \widehat{\sigma}(\mathbf{x})\Psi^{-1}(\varepsilon) + \widehat{\mu}(\mathbf{x}) \rfloor - 1$ and $l_U = \lceil -\widehat{\sigma}(\mathbf{x})\Psi^{-1}(\varepsilon) + \widehat{\mu}(\mathbf{x}) \rceil$, $\xi_l = \mathbb{P}\big(\widehat{\Gamma}(\mathbf{x}) = l\big)$, $\widetilde{\xi}_l = \widetilde{\mathbb{P}}\big(\widehat{\Gamma}(\mathbf{x}) = l\big)$, $\xi_{-j,l} = \mathbb{P}\big(\widehat{\Gamma}_{-j}(\mathbf{x}) =$

$l\big)$, $\widetilde{\xi}_{\text{-}j,l} = \widetilde{\mathbb{P}}\big(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) = l\big)$, we treat $\big|\widetilde{\omega}_\tau - \overline{\omega}_\tau\big|$ and $\big|\widetilde{v}_\tau - \bar{v}_\tau\big|$ separately. First,

$$\begin{aligned}\big|\widetilde{\omega}_\tau - \overline{\omega}_\tau\big| &\leq \sum_{l>l_U} \frac{2}{\tau+l+\gamma+1}\omega_{\tau,l} + \sum_{0\leq l<l_L} \frac{2}{\tau+l+\gamma+1}\omega_{\tau,l} + \sum_{l_L\leq l<l_U} \frac{2}{\tau+l+\gamma+1}\big|\omega_{\tau,l} - \widetilde{\omega}_{\tau,l}\big| \\ &=: S_1 + S_2 + S_3.\end{aligned}$$

Next, we turn to bound S_1 - S_3 separately.

$$\begin{aligned}S_1 &= \sum_{l>l_U} \frac{2}{\tau+l+\gamma+1}\sum_{s=1}^{\tau} \widehat{q}_{j_s}(\mathbf{x})\mathbb{P}\big(\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x}) = l\big) \leq \frac{2}{\tau+l_U+\gamma+1}\sum_{s=1}^{\tau}\sum_{l>l_U}\mathbb{P}\big(\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x}) = l\big) \\ &\leq \frac{2}{\tau+l_U+\gamma+1}\sum_{s=1}^{\tau}\big(1-\mathbb{P}\big(\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x}) \leq l_U\big)\big) \leq \frac{2\tau}{\tau+l_U+\gamma+1}\mathbb{P}\big(\widehat{\Gamma}(\mathbf{x}) > l_U\big),\end{aligned}$$

where the last inequality follows from Lemma 16. For S_2, we have

$$S_2 \leq \frac{2}{\tau+\gamma+1}\sum_{s=1}^{\tau}\sum_{0\leq l<l_L}\mathbb{P}\big(\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x}) = l\big) \leq \frac{2\tau}{\tau+\gamma+1}\mathbb{P}\big(\widehat{\Gamma}(\mathbf{x}) \leq l_L + 1\big),$$

where the last inequality follows from Lemma 16. Next, according to Theorem 1.1 in Neammanee (2005),

$$\mathbb{P}\big(\widehat{\Gamma}(\mathbf{x}) \leq l_L + 1\big) \leq \mathbb{P}\big(Z \leq \Psi^{-1}(\varepsilon)\big) + \frac{C_0}{\widehat{\sigma}^2(\mathbf{x})} = \varepsilon + \frac{C_0}{\widehat{\sigma}^2(\mathbf{x})},$$

and

$$\mathbb{P}\big(\widehat{\Gamma}(\mathbf{x}) > l_U\big) \leq \mathbb{P}\big(Z \geq \Psi^{-1}(1-\varepsilon)\big) + \frac{C_0}{\widehat{\sigma}^2(\mathbf{x})} = \varepsilon + \frac{C_0}{\widehat{\sigma}^2(\mathbf{x})},$$

where Z is a random variable following the refined normal distribution. For S_3,

$$\begin{aligned}S_3 &\leq \sum_{l=0}^{d-1}\frac{2}{\tau+l+\gamma+1}\sum_{s=1}^{\tau}\widehat{\mathbf{q}}_{j_s}(\mathbf{x})|\widetilde{\xi}_{\text{-}j,l} - \xi_{\text{-}j,l}| \leq \sum_{s=1}^{\tau}\widehat{\mathbf{q}}_{j_s}(\mathbf{x})\sum_{l=0}^{d-1}\frac{2}{\tau+l+\gamma+1}\max_{j=1,\cdots,\tau}|\widetilde{\xi}_{\text{-}j,l} - \xi_{\text{-}j,l}| \\ &\leq \min(\widehat{\mu}(\mathbf{x}),\tau)\sum_{l=0}^{d-1}\frac{2}{\tau+l+\gamma+1}\max_{s=1,\cdots,\tau}\frac{C_0}{\widehat{\sigma}^2(\mathbf{x}) - \widehat{q}_{j_s}(\mathbf{x})\big(1-\widehat{q}_{j_s}(\mathbf{x})\big)} \\ &\leq \frac{2C_0\min(\widehat{\mu}(\mathbf{x}),\tau)}{\widehat{\sigma}^2(\mathbf{x}) - 1/4}\sum_{l=1}^{d}\frac{1}{\tau+l+\gamma} = \frac{2C_0\min(\widehat{\mu}(\mathbf{x}),\tau)}{\widehat{\sigma}^2(\mathbf{x}) - 1/4}\big(H_{\tau+d+\gamma} - H_{\tau+\gamma}\big) \\ &\leq \frac{2C_0\min(\widehat{\mu}(\mathbf{x}),\tau)}{\widehat{\sigma}^2(\mathbf{x}) - 1/4}\Big(\log\big(1+\frac{d}{\tau+\gamma}\big) + 1 - \frac{1}{\tau+\gamma}\Big),\end{aligned}$$

where $H_K = \sum_{k=1}^{K} 1/k$ is the harmonic number, and the last inequality follows from the fact that $\log(K) + 1/K \leq H_K \leq \log(K) + 1$. Taken together,

$$\big|\widetilde{\omega}_\tau - \overline{\omega}_\tau\big| \leq \frac{4\tau}{\tau+\gamma+1}\big(\varepsilon + \frac{C_0}{\widehat{\sigma}^2(\mathbf{x})}\big) + \frac{2C_0\min(\widehat{\mu}(\mathbf{x}),\tau)}{\widehat{\sigma}^2(\mathbf{x}) - 1/4}\Big(\log\big(1+\frac{d}{\tau+\gamma}\big) + \frac{\tau+\gamma-1}{\tau+\gamma}\Big).$$

Similarly, for $|\widetilde{v}_\tau - \bar{v}_\tau|$,

$$|\widetilde{v}_\tau - \bar{v}_\tau| \leq \frac{2\gamma}{\tau+\gamma}\varepsilon + \frac{\gamma C_0}{\widehat{\sigma}^2(\mathbf{x})}\Big(\log\big(1+\frac{d}{\tau+\gamma-1}\big) + 1 - \frac{1}{\tau+\gamma-1}\Big).$$

This completes the proof. ■

D.4 Proof of Lemma 5

Proof With the same argument in the proof of Lemma 4, it suffices to consider

$$\big|\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x})=l)-\widetilde{\mathbb{P}}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x})=l)\big| \leq \sum_{l'=l-1}^{l}\big|\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x})\leq l')-\widetilde{\mathbb{P}}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x})\leq l')\big|.$$

Denote $I=\widehat{\sigma}(\mathbf{x})^{-1}(l+1/2-\widehat{\mu}(\mathbf{x}))$ and $I_{\text{-}j}=\widehat{\sigma}_{\text{-}j}(\mathbf{x})^{-1}(l+1/2-\widehat{\mu}_{\text{-}j}(\mathbf{x}))$, we have

$$\begin{aligned}\big|\widetilde{\mathbb{P}}(\widehat{\Gamma}(\mathbf{x})\leq l)-\widetilde{\mathbb{P}}(\widehat{\Gamma}_{\text{-}j}(\mathbf{x})\leq l)\big| &= \big|\Psi(I)-\Psi_{\text{-}j}(I_{\text{-}j})\big| \\ &\leq \big|\Phi(I)-\Phi(I_{\text{-}j})\big|+\frac{1}{6}\big|\widehat{\eta}(\mathbf{x})(1-I^2)\phi(I)-\widehat{\eta}_{\text{-}j}(\mathbf{x})(1-I_{\text{-}j}^2)\phi(I_{\text{-}j})\big| =: T_1+\frac{1}{6}T_2.\end{aligned}$$

Next, we turn to treat T_1 and T_2 separately. Without loss generalization, we assume $|I_{\text{-}j}|\geq|I|$, then

$$\begin{aligned}T_1 &\leq \Big|\int_I^{I_{\text{-}j}}\phi(\mathbf{x})d\mathbf{x}\Big| \leq |I_{\text{-}j}-I|\phi(I) \\ &= \Big(|\widehat{\sigma}^{-1}(\mathbf{x})-\widehat{\sigma}_{\text{-}j}^{-1}(\mathbf{x})|\big|l+1/2-\widehat{\mu}(\mathbf{x})\big|+\widehat{\sigma}_{\text{-}j}^{-1}(\mathbf{x})\big|\widehat{\mu}(\mathbf{x})-\widehat{\mu}_{\text{-}j}(\mathbf{x})\big|\Big)\phi(I) \\ &= \widehat{\sigma}_{\text{-}j}^{-1}(\mathbf{x})(\widehat{\sigma}(\mathbf{x})-\widehat{\sigma}_{\text{-}j}(\mathbf{x}))|I|\phi(I)+\widehat{\sigma}_{\text{-}j}^{-1}(\mathbf{x})\widehat{q}_j(\mathbf{x})\phi(I) \\ &\leq \frac{\widehat{q}_j(\mathbf{x})(1-\widehat{q}_j(\mathbf{x}))}{\widehat{\sigma}_{\text{-}j}(\mathbf{x})\big(\widehat{\sigma}_{\text{-}j}(\mathbf{x})+\widehat{\sigma}(\mathbf{x})\big)}|I|\phi(I)+\frac{1}{\sqrt{2\pi}\widehat{\sigma}_{\text{-}j}(\mathbf{x})} \leq \frac{1}{4\sqrt{2\pi}}\Big(\frac{1}{2\sqrt{e}(\widehat{\sigma}^2(\mathbf{x})-1/4)}+\frac{4}{\sqrt{\widehat{\sigma}^2(\mathbf{x})-1/4}}\Big),\end{aligned}$$

where the last inequality follows the fact that $|u|\phi(u)\leq 1/\sqrt{2e\pi}$ and $0\leq\phi(u)\leq 1/\sqrt{2\pi}$. For T_2, let $g(u)=(1-u^2)\phi(u)$, we have

$$T_2 \leq \big|\widehat{\eta}(\mathbf{x})(1-I^2)\phi(I)\big|+\big|\widehat{\eta}_{\text{-}j}(\mathbf{x})(1-I_{\text{-}j}^2)\phi(I_{\text{-}j})\big| \leq \frac{1}{\sqrt{2\pi}}\frac{\widehat{m}_3(\mathbf{x})}{(\widehat{\sigma}^2(\mathbf{x})-1/4)^{3/2}},$$

where the last inequality follows from the fact that $|g(u)|\leq 1/\sqrt{2\pi}$, and $\widehat{m}_3(\mathbf{x})=\sum_{j=1}^d\widehat{p}_j(\mathbf{x})(1-\widehat{p}_j(\mathbf{x}))(1-2\widehat{p}_j(\mathbf{x}))$. Taken together, using the same argument in the proof of Lemma 4, we have

$$\big|\widetilde{\omega}_\tau^b-\widetilde{\omega}_\tau\big| \leq \frac{1}{4\sqrt{2\pi}}\Big(\frac{1/(2\sqrt{e})}{\widehat{\sigma}^2(\mathbf{x})-1/4}+\frac{4}{\sqrt{\widehat{\sigma}^2(\mathbf{x})-1/4}}+\frac{4\widehat{m}_3(\mathbf{x})}{(\widehat{\sigma}^2(\mathbf{x})-1/4)^{3/2}}\Big)\big(\log\big(1+\frac{d}{\tau+\gamma}\big)+1\big).$$

Combining Lemma 4, the desirable result then follows. ■

D.5 Proof of Lemma 6

Proof Denote $\mathcal{K}_+=\{1\leq k\leq K:\alpha_k>0\}$. We first prove the necessity. Suppose $\mathbf{\Delta}_k^*$ is a global minimizer of $\text{Dice}_k(\cdot)$, for $k\in\mathcal{K}_+$. Then for any $\mathbf{\Delta}=(\mathbf{\Delta}_1,\cdots,\mathbf{\Delta}_K)$, we have

$$\text{mDice}_\gamma(\mathbf{\Delta}^*)=\sum_{k\in\mathcal{K}_+}\alpha_k\text{Dice}_{\gamma,k}(\mathbf{\Delta}_k^*)\leq\sum_{k\in\mathcal{K}_+}\alpha_k\text{Dice}_{\gamma,k}(\mathbf{\Delta}_k),$$

yields that $\mathbf{\Delta}^*$ is a global minimizer of $\text{mDice}_\gamma(\cdot)$. We next prove the sufficiency by contradiction. Suppose $\mathbf{\Delta}^*$ is a global minimizer of $\text{mDice}_\gamma(\cdot)$, yet there exists $k_0\in\mathcal{K}_+$ such that $\mathbf{\Delta}_{k_0}^*$ is not a minimizer of $\text{Dice}_{\gamma,k_0}(\cdot)$. Thus, there exists a segmentation rule $\widetilde{\mathbf{\Delta}}$ such that $\text{Dice}_{\gamma,k_0}(\widetilde{\mathbf{\Delta}})<\text{Dice}_{\gamma,k_0}(\mathbf{\Delta}_{k_0}^*)$, then let $\widetilde{\mathbf{\Delta}} = (\mathbf{\Delta}^*_1, \cdots, \mathbf{\Delta}^*_{k_0-1}, \widetilde{\mathbf{\Delta}}_{k_0}, \mathbf{\Delta}^*_{k_0+1}, \cdots, \mathbf{\Delta}^*_K)$

$$\text{mDice}_\gamma(\mathbf{\Delta}^*) = \sum_{k \in \mathcal{K}_+} \alpha_k \text{Dice}_{\gamma,k}(\mathbf{\Delta}^*_k) > \sum_{k \in \mathcal{K}_+} \alpha_k \text{Dice}_{\gamma,k}(\widetilde{\mathbf{\Delta}}) = \text{mDice}_\gamma(\widetilde{\mathbf{\Delta}}),$$

which leads to contradiction of that $\mathbf{\Delta}^*$ is a global minimizer of $\text{mDice}_\gamma(\cdot)$. The desirable result then follows. ■

D.6 Proof of Lemma 7

Proof Given a segmentation rule $\mathbf{\Delta}(\mathbf{X}) = (\mathbf{\Delta}_1(\mathbf{X}), \cdots, \mathbf{\Delta}_K(\mathbf{X}))$, $\text{mDice}_\gamma(\cdot)$ can be rewritten as

$$\text{mDice}_\gamma(\mathbf{\Delta}) = \sum_{k=1}^K \alpha_k \text{Dice}_{\gamma,k}(\mathbf{\Delta}_k(\mathbf{X})).$$

It is equivalent to consider the point-wise minimization on $\mathbf{X} = \mathbf{x}$:

$$\mathbf{\Delta}^*(\mathbf{x}) = \underset{\mathbf{V} \in \{0,1\}^{d \times K}}{\text{argmax}} \ \text{mDice}_\gamma(\mathbf{V}|\mathbf{x}), \quad \text{s.t.} \sum_{k=1}^K \mathbf{V}_k = \mathbf{1}_d, \quad \text{mDice}_\gamma(\mathbf{V}|\mathbf{x}) = \sum_{k=1}^K \alpha_k \text{Dice}_{\gamma,k}(\mathbf{V}_k|\mathbf{x}),$$

where $\mathbf{V}_k$ is the k-th column of $\mathbf{V}$. According to (25), we have

$$\begin{aligned}\text{mDice}_\gamma(\mathbf{V}|\mathbf{x}) &= \sum_{k=1}^K \alpha_k \sum_{j \in I(\mathbf{V}_k)} \sum_{\substack{\mathbf{y}_{-j,k} \in \{0,1\}^{d-1} \\ y_{j,k}=1}} \frac{2\mathbb{P}(\mathbf{Y}_k = \mathbf{y}_k|\mathbf{x})}{\tau_k + \|\mathbf{y}_k\|_1 + \gamma} + \sum_{k=1}^K \alpha_k \mathbb{E}\Big(\frac{\gamma}{\|\mathbf{Y}_{\cdot k}\|_1 + \tau_k + \gamma}\Big) \\ &= \sum_{k=1}^K \sum_{j=1}^d R_{jk}(\tau_k) v_{jk} + \sum_{k=1}^K \alpha_k \bar{v}(\tau_k),\end{aligned}$$

where $v_{jk} \in \{0,1\}$ is the segmentation indicator of the j-th feature under the class-k, and $R_{jk}(\cdot)$ is a reward function defined as:

$$R_{jk}(\tau_k) = \alpha_k \sum_{\substack{\mathbf{y}_{-j,k} \in \{0,1\}^{d-1} \\ y_{j,k}=1}} \frac{2\mathbb{P}(\mathbf{Y}_k = \mathbf{y}_k|\mathbf{x})}{\tau_k + \|\mathbf{y}_k\|_1}.$$

Now, suppose the optimal volume function $\boldsymbol{\tau}^*(\mathbf{x}) = (\tau^*_1(\mathbf{x}), \cdots, \tau^*_K(\mathbf{x}))^\intercal$ is given, then $\bar{v}(\tau^*_k)$ becomes a constant, and the point-wise maximization on mDice_γ is equivalent to:

$$\begin{aligned}&\max_{\mathbf{V} \in \{0,1\}^{d \times K}} && \sum_{k=1}^K \sum_{j=1}^d R^*_{jk} v_{jk}, \\ &\text{subject to} && \sum_{j=1}^d v_{jk} = \tau^*_k(\mathbf{x}), \quad \sum_{k=1}^K v_{jk} = 1, \quad \text{for } k = 1, \cdots, K; \quad \text{for } j = 1, \cdots, d, \end{aligned} \tag{28}$$

where $R^*_{jk} = R_{jk}(\tau^*_k(\mathbf{x}))$ is the reward under $\tau^*_k(\mathbf{x})$. Note that (28) is the formulation for the assignment problem (Kuhn, 1955). This completes the proof. ■

D.7 Proof of Lemma 9

Proof For simplicity, we construct a counter example based on $\gamma = 0$ and $d = 2$, that is, $\mathbf{Y} = (Y_1, Y_2)^\intercal$ and $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}))^\intercal$. Without loss generality, we assume $p_1(\mathbf{x}) > p_2(\mathbf{x})$.

First, we derive the Bayes rule in Theorem 1 for this case. Note that it suffices to compare the scores for $\tau = 1$ ($\mathbf{v} = (1,0)^\intercal$) and $\tau = 2$ ($\mathbf{v} = (1,1)^\intercal$).

$$\operatorname{Dice}((1,0)^\intercal|\mathbf{x}) = p_1(\mathbf{x}) + \frac{1}{3}p_1(\mathbf{x})p_2(\mathbf{x}), \quad \operatorname{Dice}((1,1)^\intercal|\mathbf{x}) = \frac{2}{3}(p_1(\mathbf{x}) + p_2(\mathbf{x})) - \frac{1}{3}p_1(\mathbf{x})p_2(\mathbf{x}).$$

Therefore, the Bayes rule for Dice optimal segmentation is:

$$\boldsymbol{\delta}^*(\mathbf{x}) = (1,0)^\intercal, \text{ if } \frac{1}{2}p_1(\mathbf{x}) - p_2(\mathbf{x}) + p_1(\mathbf{x})p_2(\mathbf{x}) \geq 0, \quad \boldsymbol{\delta}^*(\mathbf{x}) = (1,1)^\intercal, \text{ otherwise.}$$

Now, we check the Dice-calibrated (Fisher consistency) for classification-calibrated losses (CCL). Let $p_1(\mathbf{x}) = 0.45$ and $p_2(\mathbf{x}) = 0.44$, then $\widetilde{\boldsymbol{\delta}}(\mathbf{x}) = \mathbf{1}(\mathbf{p}(\mathbf{x}) \geq 0.5) = (0,0)^\intercal \neq (1,1)^\intercal = \boldsymbol{\delta}^*(\mathbf{x})$, where the first equality follows from the definition of a classification-calibrated loss. Therefore, $\operatorname{Dice}(\widetilde{\boldsymbol{\delta}}) < \operatorname{Dice}(\boldsymbol{\delta}^*)$ yields that a classification-calibrated loss with thresholding at 0.5 is not Dice-calibrated.

Next, we check the Dice calibration for soft-Dice loss. Again, it suffices to consider the point-wise minimization given $\mathbf{X} = \mathbf{x}$, and its gradients with respect to $c_j = q_j(\mathbf{x}) \in [0,1]$ for $j = 1, 2$:

$$\begin{aligned}
\frac{\partial \mathbb{E}\big(l_{\text{SoftD}}(\mathbf{Y}, \mathbf{c}) \mid \mathbf{X} = \mathbf{x}\big)}{\partial c_1} &= -2\mathbb{E}\Big(\frac{Y_1(Y_1 + Y_2 + c_1 + c_2) - (Y_1 c_1 + Y_2 c_2)}{(Y_1 + Y_2 + c_1 + c_2)^2} \mid \mathbf{X} = \mathbf{x}\Big) \\
&= -2\Big(\frac{2p_1(\mathbf{x})p_1(\mathbf{x})}{(2 + c_1 + c_2)^2} + \frac{p_1(\mathbf{x})(1 - p_2(\mathbf{x})) + (p_1(\mathbf{x}) - p_2(\mathbf{x}))c_2}{(1 + c_1 + c_2)^2}\Big), \\
\frac{\partial \mathbb{E}\big(l_{\text{SoftD}}(\mathbf{Y}, \mathbf{c}) \mid \mathbf{X} = \mathbf{x}\big)}{\partial c_2} &= -2\Big(\frac{2p_1(\mathbf{x})p_1(\mathbf{x})}{(2 + c_1 + c_2)^2} + \frac{p_2(\mathbf{x})(1 - p_1(\mathbf{x})) + (p_2(\mathbf{x}) - p_1(\mathbf{x}))c_1}{(1 + c_1 + c_2)^2}\Big).
\end{aligned}$$

Accordingly, $\partial \mathbb{E}\big(l_{\text{SoftD}}(\mathbf{Y}, \mathbf{c}) \mid \mathbf{X} = \mathbf{x}\big)/\partial c_1 \leq 0$ for any $c_2 \in [0,1]$, and $\partial \mathbb{E}\big(l_{\text{SoftD}}(\mathbf{Y}, \mathbf{c}) \mid \mathbf{X} = \mathbf{x}\big)/\partial c_2 \leq 0$ when $2p_2(\mathbf{x}) - p_1(\mathbf{x}) - p_1(\mathbf{x})p_2(\mathbf{x}) \geq 0$. For example, let $p_1(\mathbf{x}) = 0.45$ and $p_2(\mathbf{x}) = 0.4$, then $\widetilde{\boldsymbol{\delta}}(\mathbf{x}) = (1,1)^\intercal \neq (1,0)^\intercal = \boldsymbol{\delta}^*(\mathbf{x})$. Therefore, $\operatorname{Dice}(\widetilde{\boldsymbol{\delta}}) < \operatorname{Dice}(\boldsymbol{\delta}^*)$ yields that the soft-Dice loss with thresholding at 0.5 is not Dice-calibrated. This completes the proof. ■

D.8 Proof of Lemma 10

Proof By the definition of a strictly proper loss and the formulation in (24), we have $\widehat{\mathbf{q}}(\mathbf{x}) = \mathbf{p}(\mathbf{x}) = \big(\mathbb{P}(Y_1 = 1|\mathbf{X} = \mathbf{x}), \cdots, \mathbb{P}(Y_d = 1|\mathbf{X} = \mathbf{x})\big)^\intercal$. Then the estimation of $\widehat{\tau}(\mathbf{x})$ and $\widehat{\boldsymbol{\delta}}(\mathbf{x})$ agrees with the definition of $\tau^*(\mathbf{x})$ and $\boldsymbol{\delta}^*(\mathbf{x})$. This completes the proof. ■

D.9 Proof of Theorem 11

Proof First, we consider point-wise approximation of the Dice metric under probabilities $\mathbf{p}$ and $\widehat{\mathbf{q}}$. For any $\boldsymbol{\delta}$, such that $\delta_j(\mathbf{x}) = 1$ for $j = j_1, \cdots, j_\tau$, and $\delta_j(\mathbf{x}) = 0$ otherwise. Define

$$
\begin{aligned}
\widehat{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}(\mathbf{x})|\mathbf{X}=\mathbf{x}) &:= \sum_{s=1}^{\tau}\sum_{l=0}^{d-1} \frac{2\widehat{q}_{j_s}(\mathbf{x})\mathbb{P}\big(\widehat{\Gamma}_{\text{-}j}(\mathbf{x}) = l\big)}{\tau+l+\gamma+1} + \sum_{l=0}^{d} \frac{\gamma\mathbb{P}\big(\widehat{\Gamma}(\mathbf{X}) = l\big)}{\tau+l+\gamma} \\
&= \sum_{s=1}^{\tau} 2\widehat{q}_{j_s}(\mathbf{x})\mathbb{E}\big(\frac{1}{\tau+\gamma+1+\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x})}\big) + \gamma\mathbb{E}\big(\frac{1}{\tau+\gamma+\widehat{\Gamma}(\mathbf{x})}\big) \\
\mathrm{Dice}_\gamma(\boldsymbol{\delta}(\mathbf{x})|\mathbf{X}=\mathbf{x}) &:= \sum_{s=1}^{\tau}\sum_{l=0}^{d-1} \frac{2p_{j_s}(\mathbf{x})\mathbb{P}\big(\Gamma_{\text{-}j}(\mathbf{x}) = l\big)}{\tau+l+\gamma+1} + \sum_{l=0}^{d} \frac{\gamma\mathbb{P}\big(\Gamma(\mathbf{X}) = l\big)}{\tau+l+\gamma} \\
&= \sum_{s=1}^{\tau} 2p_{j_s}(\mathbf{x})\mathbb{E}\big(\frac{1}{\tau+\gamma+1+\Gamma_{\text{-}j_s}(\mathbf{x})}\big) + \gamma\mathbb{E}\big(\frac{1}{\tau+\gamma+\Gamma(\mathbf{x})}\big).
\end{aligned}
$$

Now, we have

$$
\begin{aligned}
&\big|\widehat{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}(\mathbf{x})|\mathbf{X}=\mathbf{x}) - \mathrm{Dice}_\gamma(\boldsymbol{\delta}(\mathbf{x})|\mathbf{X}=\mathbf{x})\big| \\
&\leq \Big|2\sum_{s=1}^{\tau}\Big(\widehat{q}_{j_s}(\mathbf{x})\mathbb{E}\big(\frac{1}{\tau+\gamma+1+\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x})}\big) - p_{j_s}(\mathbf{x})\mathbb{E}\big(\frac{1}{\tau+\gamma+1+\Gamma_{\text{-}j_s}(\mathbf{x})}\big)\Big)\Big| \\
&\qquad + \gamma\Big|\mathbb{E}\big(\frac{1}{\tau+\gamma+\widehat{\Gamma}(\mathbf{x})}\big) - \mathbb{E}\big(\frac{1}{\tau+\gamma+\Gamma(\mathbf{x})}\big)\Big| \\
&\leq \Big|2\sum_{s=1}^{\tau}\widehat{q}_{j_s}(\mathbf{x})\Big(\mathbb{E}\big(\frac{1}{\tau+\gamma+1+\widehat{\Gamma}_{\text{-}j_s}(\mathbf{x})}\big) - \mathbb{E}\big(\frac{1}{\tau+\gamma+1+\Gamma_{\text{-}j_s}(\mathbf{x})}\big)\Big)\Big| \\
&\qquad + \Big|2\sum_{s=1}^{\tau}\big(\widehat{q}_{j_s}(\mathbf{x}) - p_{j_s}(\mathbf{x})\big)\mathbb{E}\big(\frac{1}{\tau+\gamma+1+\Gamma_{\text{-}j_s}(\mathbf{x})}\big)\Big)\Big| + \gamma\Big|\mathbb{E}\big(\frac{1}{\tau+\gamma+\widehat{\Gamma}(\mathbf{x})}\big) - \mathbb{E}\big(\frac{1}{\tau+\gamma+\Gamma(\mathbf{x})}\big)\Big| \\
&\leq 2\sum_{s=1}^{\tau}\Big|\frac{\mathbb{E}\big(\Gamma_{\text{-}j_s}(\mathbf{x}) - \widehat{\Gamma}_{\text{-}j_s}(\mathbf{x})\big)}{(\tau+\gamma+1)^2}\Big| + 2\sum_{s=1}^{\tau}\frac{\big|\widehat{q}_{j_s}(\mathbf{x}) - p_{j_s}(\mathbf{x})\big|}{\tau+\gamma+1} + \gamma\Big|\frac{\mathbb{E}\big(\Gamma(\mathbf{x}) - \widehat{\Gamma}(\mathbf{x})\big)}{(\tau+\gamma)^2}\Big| \\
&\leq 2\sum_{s=1}^{\tau}\frac{\big|\|\widehat{\mathbf{q}}(\mathbf{x})\|_1 - \|\mathbf{p}(\mathbf{x})\|_1\big| + \big|\widehat{q}_{j_s}(\mathbf{x}) - p_{j_s}(\mathbf{x})\big|}{(\tau+\gamma+1)^2} + 2\sum_{s=1}^{\tau}\frac{\big|\widehat{q}_{j_s}(\mathbf{x}) - p_{j_s}(\mathbf{x})\big|}{\tau+\gamma+1} + \gamma\frac{\big|\|\widehat{\mathbf{q}}(\mathbf{x})\|_1 - \|\mathbf{p}(\mathbf{x})\|_1\big|}{(\tau+\gamma)^2} \\
&\leq \big(\frac{3}{2(1+\gamma)} + c_1\big)\|\widehat{\mathbf{q}}(\mathbf{x}) - \mathbf{p}(\mathbf{x})\|_1,
\end{aligned}
$$

where the second inequality follows from the triangle inequality, $c_1 = 0$ if $\gamma = 0$, $c_1 = 1/\gamma$ if $\gamma > 0$. Therefore,

$$
\begin{aligned}
\mathrm{Dice}_\gamma(\boldsymbol{\delta}^*) - \mathrm{Dice}_\gamma(\widehat{\boldsymbol{\delta}}) &= \mathrm{Dice}_\gamma(\boldsymbol{\delta}^*) - \widehat{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}^*) + \widehat{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}^*) - \widehat{\mathrm{Dice}}_\gamma(\widehat{\boldsymbol{\delta}}) + \widehat{\mathrm{Dice}}_\gamma(\widehat{\boldsymbol{\delta}}) - \mathrm{Dice}_\gamma(\widehat{\boldsymbol{\delta}}) \\
&\leq \mathrm{Dice}_\gamma(\boldsymbol{\delta}^*) - \widehat{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}^*) + \widehat{\mathrm{Dice}}_\gamma(\widehat{\boldsymbol{\delta}}) - \mathrm{Dice}_\gamma(\widehat{\boldsymbol{\delta}}) \\
&\leq \mathbb{E}_{\mathbf{X}}\big|\widehat{\mathrm{Dice}}_\gamma(\boldsymbol{\delta}^*(\mathbf{X})|\mathbf{X}) - \mathrm{Dice}_\gamma(\boldsymbol{\delta}^*(\mathbf{X})|\mathbf{X})\big| + \mathbb{E}_{\mathbf{X}}\big|\widehat{\mathrm{Dice}}_\gamma(\widehat{\boldsymbol{\delta}}(\mathbf{X})|\mathbf{X}) - \mathrm{Dice}_\gamma(\widehat{\boldsymbol{\delta}}(\mathbf{X})|\mathbf{X})\big| \\
&\leq \big(\frac{3}{1+\gamma} + 2c_1\big)\mathbb{E}\|\widehat{\mathbf{q}}(\mathbf{X}) - \mathbf{p}(\mathbf{X})\|_1,
\end{aligned}
$$

where the first inequality follows from the definition of $\widehat{\boldsymbol{\delta}}$ such that $\widehat{\text{Dice}}_\gamma(\boldsymbol{\delta}^*) - \widehat{\text{Dice}}_\gamma(\widehat{\boldsymbol{\delta}}) \leq 0$. This completes the proof. ■

D.10 Proof of Corollary 12

Proof According to the Pinsker's inequality, we have

$$\begin{aligned}\mathbb{E}_{\mathbf{X}}\|\widehat{\mathbf{q}}(\mathbf{X}) - \mathbf{p}(\mathbf{X})\|_1 &= \sum_{j=1}^d \mathbb{E}_{\mathbf{X}}|\widehat{q}_j(\mathbf{X}) - p_j(\mathbf{X})| \leq \sum_{j=1}^d \mathbb{E}_{\mathbf{X}}\sqrt{\frac{1}{2}\text{KL}\big(\mathbb{P}(Y_j|\mathbf{X}), \widehat{\mathbb{P}}(Y_j|\mathbf{X})\big)} \\ &\leq \sqrt{\frac{d}{2}\sum_{j=1}^d \mathbb{E}_{\mathbf{X}}\text{KL}\big(\mathbb{P}(Y_j|\mathbf{X}), \widehat{\mathbb{P}}(Y_j|\mathbf{X})\big)} = \sqrt{\frac{d}{2}}\sqrt{\mathbb{E}\Big(l_{\text{CE}}\big(\mathbf{Y}, \widehat{\mathbf{q}}(\mathbf{X})\big)\Big) - \mathbb{E}\Big(l_{\text{CE}}\big(\mathbf{Y}, \mathbf{p}(\mathbf{X})\big)\Big)},\end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality and the Jensen's inequality, and $\text{KL}\big(\mathbb{P}(Y_j|\mathbf{x}), \widehat{\mathbb{P}}(Y_j|\mathbf{x})\big) := p_j(\mathbf{x})\log\big(p_j(\mathbf{x})/\widehat{q}_j(\mathbf{x})\big) + (1-p_j(\mathbf{x}))\log((1-p_j(\mathbf{x}))/(1-\widehat{q}_j(\mathbf{x})))$ is the KL divergence between $\mathbb{P}(Y_j|\mathbf{x})$ under $\mathbf{p}$ and $\widehat{\mathbb{P}}(Y_j|\mathbf{x})$ under $\widehat{\mathbf{q}}$. The desirable result then follows by combining (19) in Theorem 11. This completes the proof. ■

D.11 Proof of Lemma 13

Proof Denote $\mathbf{y}_I = (y_j : j \in I)^\intercal$, $\mathbf{y}_{\text{-}I} = (y_j : j \notin I)^\intercal$, and let $I(\mathbf{v}) = I(\boldsymbol{\delta}(\mathbf{x})) = \{j : v_j = 1\}$ be a segmentation index set, and $\|\mathbf{v}\|_1 = \tau$. Again, consider the point-wise maximization:

$$\boldsymbol{\delta}^*(\mathbf{x}) = \underset{\mathbf{v}\in\{0,1\}^d}{\text{argmax}}\ \text{IoU}_\gamma(\mathbf{v}|\mathbf{x}),$$

where $\text{IoU}_\gamma(\mathbf{v}|\mathbf{x})$ is defined as

$$\begin{aligned}\text{IoU}_\gamma(\mathbf{v}|\mathbf{x}) &= \mathbb{E}\Big(\frac{\|\mathbf{Y}_{I(\mathbf{v})}\|_1 + \gamma}{\|\mathbf{Y}_{\text{-}I(\mathbf{v})}\|_1 + \|\mathbf{v}\|_1 + \gamma}\Big|\mathbf{X} = \mathbf{x}\Big) \\ &= \mathbb{E}\Big(\frac{\|\mathbf{Y}_{I(\mathbf{v})}\|_1}{\|\mathbf{Y}_{\text{-}I(\mathbf{v})}\|_1 + \tau + \gamma}\Big|\mathbf{X} = \mathbf{x}\Big) + \mathbb{E}\Big(\frac{\gamma}{\|\mathbf{Y}_{\text{-}I(\mathbf{v})}\|_1 + \tau + \gamma}\Big|\mathbf{X} = \mathbf{x}\Big) \\ &= \Big(\mathbb{E}(\|\mathbf{Y}_{I(\mathbf{v})}\|_1|\mathbf{X} = \mathbf{x}) + \gamma\Big)\mathbb{E}\Big(\frac{1}{\|\mathbf{Y}_{\text{-}I(\mathbf{v})}\|_1 + \tau + \gamma}\Big|\mathbf{X} = \mathbf{x}\Big),\end{aligned}$$

where the last equality follows from the fact that $\mathbf{Y}_{I(\mathbf{v})} \perp \mathbf{Y}_{\text{-}I(\mathbf{v})} \mid \mathbf{X}$. Fix $\tau = \|\mathbf{v}\|_1$. The first term is maximized at $\mathbf{v}^* = (v_1^*, \cdots, v_d^*)$ with

$$v_j^* = \begin{cases} 1 & \text{if } p_j(\mathbf{x}) \text{ ranks top } \tau, \\ 0 & \text{otherwise,} \end{cases}$$

and the maximum value is $\sum_{j\in J_\tau(\mathbf{x})} p_j(\mathbf{x}) + \gamma$. The second term is

$$\mathbb{E}\Big(\frac{1}{\|\mathbf{Y}_{\text{-}I(\mathbf{v})}\|_1 + \tau + \gamma}\Big|\mathbf{X} = \mathbf{x}\Big) = \mathbb{E}\Big(\frac{1}{\Gamma_{\text{-}I(\mathbf{v})}(\mathbf{x}) + \tau + \gamma}\Big).$$

Given $I(\mathbf{v}) \neq I(\mathbf{v}^*)$, we have $\Gamma_{\text{-}I(\mathbf{v})}(\mathbf{x}) = \sum_{j=1}^{d-\tau} B_j$ and $\Gamma_{\text{-}I(\mathbf{v}^*)}(\mathbf{x}) = \sum_{j=1}^{d-\tau} B_j^*$, where B_j and B_j^* are independent Bernoulli variables with success probability $p_j \geq p_j^*$, respectively, for $j = 1, \cdots, d-\tau$. By Theorem 2.2.9 of Belzunce et al. (2015), $\Gamma_{\text{-}I(\mathbf{v})}(\mathbf{x})$ is stochastically greater than $\Gamma_{\text{-}I(\mathbf{v}^*)}(\mathbf{x})$, namely

$$\mathbb{P}(\Gamma_{\text{-}I(\mathbf{v})}(\mathbf{x}) \leq \varsigma) \leq \mathbb{P}(\Gamma_{\text{-}I(\mathbf{v}^*)}(\mathbf{x}) \leq \varsigma)$$

for any $\varsigma \in \mathbb{R}$. Thus,

$$\begin{aligned}\mathbb{E}\Big(\frac{1}{\Gamma_{\text{-}I(\mathbf{v})}(\mathbf{x}) + \tau + \gamma}\Big) &= \int_0^\infty \mathbb{P}\Big(\frac{1}{\Gamma_{\text{-}I(\mathbf{v})}(\mathbf{x}) + \tau + \gamma} \geq \varsigma\Big) d\varsigma \\ &\leq \int_0^\infty \mathbb{P}\Big(\frac{1}{\Gamma_{\text{-}I(\mathbf{v}^*)}(\mathbf{x}) + \tau + \gamma} \geq \varsigma\Big) d\varsigma = \mathbb{E}\Big(\frac{1}{\Gamma_{\text{-}I(\mathbf{v}^*)}(\mathbf{x}) + \tau + \gamma}\Big).\end{aligned}$$

As a result, the second term is also maximized at $\mathbf{v}^*$. Therefore, $\mathbf{v}^*$ maximizes $\text{IoU}_\gamma(\mathbf{v}|\mathbf{x})$ for a fixed τ. The desired result follows. ■

D.12 Proof of Lemma 14

Proof Without loss of generality, assuming $\widehat{q}_1(\mathbf{x}) \geq \cdots \geq \widehat{q}_d(\mathbf{x})$ are fixed, then for any $\tau' > \tau$,

$$\begin{aligned}&\varpi_\tau(\mathbf{x}) - \varpi_{\tau'}(\mathbf{x}) \\ &= \Big(\sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) + \gamma\Big)\mathbb{E}\Big(\frac{1}{\widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})} + \tau + \gamma}\Big) - \Big(\sum_{j=1}^{\tau'} \widehat{q}_j(\mathbf{x}) + \gamma\Big)\mathbb{E}\Big(\frac{1}{\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \tau' + \gamma}\Big) \\ &= \Big(\sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) + \gamma\Big)\mathbb{E}\Big(\frac{1}{\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \sum_{j=\tau+1}^{\tau'} \widehat{B}_j(\mathbf{x}) + \tau + \gamma}\Big) - \Big(\sum_{j=1}^{\tau'} \widehat{q}_j(\mathbf{x}) + \gamma\Big)\mathbb{E}\Big(\frac{1}{\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \tau' + \gamma}\Big) \\ &= \mathbb{E}\Big(\frac{(\sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) + \gamma)(\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \tau' + \gamma) - (\sum_{j=1}^{\tau'} \widehat{q}_j(\mathbf{x}) + \gamma)(\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \sum_{j=\tau+1}^{\tau'} \widehat{B}_j(\mathbf{x}) + \tau + \gamma)}{(\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \sum_{j=\tau+1}^{\tau'} \widehat{B}_j(\mathbf{x}) + \tau + \gamma)(\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \tau' + \gamma)}\Big) \\ &= \mathbb{E}\Big(\frac{(\sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) + \gamma)(\tau' - \tau - \sum_{j=\tau+1}^{\tau'} \widehat{B}_j(\mathbf{x}))}{(\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \sum_{j=\tau+1}^{\tau'} \widehat{B}_j(\mathbf{x}) + \tau + \gamma)(\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \tau' + \gamma)}\Big) - \mathbb{E}\Big(\frac{\sum_{j=\tau+1}^{\tau'} \widehat{q}_j(\mathbf{x})}{\widehat{\Gamma}_{\text{-}J_{\tau'}(\mathbf{x})} + \tau' + \gamma}\Big) \\ &\geq \frac{(\sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) + \gamma)(\tau' - \tau - \sum_{j=\tau+1}^{\tau'} \widehat{q}_j(\mathbf{x}))}{((d-\tau')\widehat{q}_{\tau+1}(\mathbf{x}) + \tau' + \gamma)^2} - \frac{\sum_{j=\tau+1}^{\tau'} \widehat{q}_j(\mathbf{x})}{\tau' + \gamma} \\ &\geq \frac{(\sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) + \gamma)(\tau' - \tau)(1 - \widehat{q}_{\tau+1}(\mathbf{x}))}{((d-\tau')\widehat{q}_{\tau+1}(\mathbf{x}) + \tau' + \gamma)^2} - \frac{(\tau' - \tau)\widehat{q}_{\tau+1}(\mathbf{x})}{\tau' + \gamma} \geq 0,\end{aligned}$$

whenever

$$\begin{aligned}\sum_{j=1}^{\tau} \widehat{q}_j(\mathbf{x}) + \gamma &\geq \frac{\widehat{q}_{\tau+1}(\mathbf{x})}{1 - \widehat{q}_{\tau+1}(\mathbf{x})} \max_{\tau' > \tau} \frac{((d-\tau')\widehat{q}_{\tau+1}(\mathbf{x}) + \tau' + \gamma)^2}{\tau' + \gamma} \\ &= \frac{\widehat{q}_{\tau+1}(\mathbf{x})}{1 - \widehat{q}_{\tau+1}(\mathbf{x})} \max\Big(d + \gamma, \frac{((d-\tau)\widehat{q}_{\tau+1}(\mathbf{x}) + \tau + \gamma)^2}{\tau + \gamma}\Big).\end{aligned}$$

This completes the proof. ■

D.13 Proof of Theorem 15

Proof Similar to the proof of Theorem 11, we consider point-wise approximation of IoU metric under $\mathbf{p}$ and $\widehat{\mathbf{q}}$. For any $\boldsymbol{\delta}$ such that $\delta_j(\mathbf{x}) = 1$ for $j \in J_\tau(\mathbf{x})$ and $\delta_j(\mathbf{x}) = 0$ otherwise. Define

$$\begin{aligned}
\widehat{\text{IoU}}_\gamma(\boldsymbol{\delta}(\mathbf{x})|\mathbf{X}=\mathbf{x}) &:= \Big(\sum_{s=1}^{\tau} \widehat{q}_{j_s}(\mathbf{x}) + \gamma\Big) \sum_{l=0}^{d-\tau} \frac{\mathbb{P}\big(\widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = l\big)}{\tau + l + \gamma} \\
&= \Big(\sum_{s=1}^{\tau} \widehat{q}_{j_s}(\mathbf{x}) + \gamma\Big) \mathbb{E}\big(\frac{1}{\tau + \gamma + \widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big), \\
\text{IoU}_\gamma(\boldsymbol{\delta}(\mathbf{x})|\mathbf{X}=\mathbf{x}) &:= \Big(\sum_{s=1}^{\tau} p_j(\mathbf{x}) + \gamma\Big) \sum_{l=0}^{d-\tau} \frac{\mathbb{P}\big(\Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) = l\big)}{\tau + l + \gamma} \\
&= \Big(\sum_{s=1}^{\tau} p_j(\mathbf{x}) + \gamma\Big) \mathbb{E}\big(\frac{1}{\tau + \gamma + \Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big).
\end{aligned}$$

We have

$$\begin{aligned}
&\big|\widehat{\text{IoU}}_\gamma(\boldsymbol{\delta}(\mathbf{x})|\mathbf{X}=\mathbf{x}) - \text{IoU}_\gamma(\boldsymbol{\delta}(\mathbf{x})|\mathbf{X}=\mathbf{x})\big| \\
&\leq \Big|\sum_{s=1}^{\tau} \Big(\widehat{q}_{j_s}(\mathbf{x})\mathbb{E}\big(\frac{1}{\tau + \gamma + \widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big) - p_{j_s}(\mathbf{x})\mathbb{E}\big(\frac{1}{\tau + \gamma + \Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big)\Big)\Big| \\
&\quad + \gamma\Big|\mathbb{E}\big(\frac{1}{\tau + \gamma + \widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big) - \mathbb{E}\big(\frac{1}{\tau + \gamma + \Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big)\Big| \\
&\leq \Big|\sum_{s=1}^{\tau} \widehat{q}_{j_s}(\mathbf{x})\Big(\mathbb{E}\big(\frac{1}{\tau + \gamma + 1 + \widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big) - \mathbb{E}\big(\frac{1}{\tau + \gamma + 1 + \Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big)\Big)\Big| \\
&\quad + \Big|\sum_{s=1}^{\tau} (\widehat{q}_{j_s}(\mathbf{x}) - p_{j_s}(\mathbf{x}))\mathbb{E}\big(\frac{1}{\tau + \gamma + 1 + \Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big)\big)\Big| \\
&\quad + \gamma\Big|\mathbb{E}\big(\frac{1}{\tau + \gamma + \widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big) - \mathbb{E}\big(\frac{1}{\tau + \gamma + \Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})}\big)\Big| \\
&\leq \sum_{s=1}^{\tau} \Big|\frac{\mathbb{E}\big(\Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) - \widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})\big)}{(\tau + \gamma + 1)^2}\Big| + \sum_{s=1}^{\tau} \frac{\big|\widehat{q}_{j_s}(\mathbf{x}) - p_{j_s}(\mathbf{x})\big|}{\tau + \gamma + 1} + \gamma\Big|\frac{\mathbb{E}\big(\Gamma_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x}) - \widehat{\Gamma}_{\text{-}J_\tau(\mathbf{x})}(\mathbf{x})\big)}{(\tau + \gamma)^2}\Big| \\
&\leq \sum_{s=1}^{\tau} \frac{\|\widehat{\mathbf{q}}(\mathbf{x}) - \mathbf{p}(\mathbf{x})\|_1}{(\tau + \gamma + 1)^2} + \sum_{s=1}^{\tau} \frac{\big|\widehat{q}_{j_s}(\mathbf{x}) - p_{j_s}(\mathbf{x})\big|}{\tau + \gamma + 1} + \gamma\frac{\|\widehat{\mathbf{q}}(\mathbf{x}) - \mathbf{p}(\mathbf{x})\|_1}{(\tau + \gamma)^2} \\
&\leq C_2\|\widehat{\mathbf{q}}(\mathbf{x}) - \mathbf{p}(\mathbf{x})\|_1,
\end{aligned}$$

where C_2 is a constant that may depend on γ. Then the rest of the proof follows from the same arguments for Theorems 11 and 12 with Dice replaced by IoU, and is omitted. ■

Appendix E. Auxiliary Lemmas

Lemma 16 *Suppose $\Gamma = \sum_{j=1}^{d} B_j$ is a Poisson binomial random variable, and $B_j (j = 1, \cdots, d)$ are independent Bernoulli trials with success probabilities $p_1, \cdots, p_d$. Then, for any $j = 1, \cdots, d$, we* *have*

$$\mathbb{P}\big(\Gamma_{\text{-}j} \leq l-1\big) \leq \mathbb{P}\big(\Gamma \leq l\big) \leq \mathbb{P}\big(\Gamma_{\text{-}j} \leq l\big),$$

where $\Gamma_{\text{-}j} = \sum_{j' \neq j} B_{j'}$.

Proof Note that $\Gamma = \Gamma_{\text{-}j} + B_j$, then

$$\begin{aligned}\mathbb{P}(\Gamma \leq l) &= \mathbb{P}(\Gamma_{\text{-}j} + B_j \leq l) = \mathbb{P}(\Gamma_{\text{-}j} \leq l \mid B_j = 0)(1-p_j) + \mathbb{P}(\Gamma_{\text{-}j} \leq l-1 \mid B_j = 1)p_j \\ &= \mathbb{P}(\Gamma_{\text{-}j} \leq l)(1-p_j) + \mathbb{P}(\Gamma_{\text{-}j} \leq l-1)p_j,\end{aligned}$$

where the last equality follows from the fact that $B_j \perp B_{j'}$ for $j \neq j'$. The desirable result then follows since $\mathbb{P}(\Gamma_{\text{-}j} \leq l) \geq \mathbb{P}(\Gamma_{\text{-}j} \leq l-1)$. ■

Lemma 17 *Suppose $\Gamma = \sum_{j=1}^{d} B_j$ is a Poisson binomial random variable, and $B_j (j = 1, \cdots, d)$ are independent Bernoulli trials with success probabilities $p_1 \geq p_2 \geq \cdots \geq p_d$. Then, for an arbitrary positive non-decreasing sequence ζ_l $(l = 0, \cdots, d-1)$,*

$$Q_j = \sum_{l=0}^{d-1} \frac{1}{\zeta_l} \mathbb{P}(\Gamma_{\text{-}j} = l)$$

is a non-increasing function with respect to $j = 1, \cdots, d$.

Proof Note that for any $j' \neq j$, $\Gamma_{\text{-}j} = \sum_{i \neq j} B_i = \sum_{i \neq j, j'} B_i + B_{j'} =: \Gamma_{\text{-}jj'} + B_{j'}$. Denote $\xi_{\text{-}jj',l} = \mathbb{P}\big(\Gamma_{\text{-}jj'} = l\big)$, we have

$$\mathbb{P}\big(\Gamma_{\text{-}j} = l\big) = (1-p_{j'})\xi_{\text{-}jj',l} + p_{j'}\xi_{\text{-}jj',l-1}.$$

Then,

$$\begin{aligned}Q_j - Q_{j'} &= \sum_{l=0}^{d} \frac{1}{\zeta_l}\big((1-p_{j'})\xi_{\text{-}jj',l} + p_{j'}\xi_{\text{-}jj',l-1} - (1-p_j)\xi_{\text{-}jj',l} - p_j\xi_{\text{-}jj',l-1}\big) \\ &= \sum_{l=0}^{d} \frac{1}{\zeta_l}(\xi_{\text{-}jj',l} - \xi_{\text{-}jj',l-1}) = (p_j - p_{j'})\Big(\sum_{l=0}^{d} \frac{1}{\zeta_l}\xi_{\text{-}jj',l} - \sum_{l=0}^{d} \frac{1}{\zeta_l}\xi_{\text{-}jj',l-1}\Big) \\ &= (p_j - p_{j'})\Big(\sum_{l=0}^{d}\big(\frac{1}{\zeta_l} - \frac{1}{\zeta_{l+1}}\big)\xi_{\text{-}jj',l}\Big),\end{aligned}$$

where the first equality follows from the fact that $\mathbb{P}(\Gamma_{\text{-}j} = d) = 0$, and the last equality follows from the fact that $\xi_{\text{-}jj',l} = 0$ for $l < 0$. Hence, we have $Q_j - Q_{j'}$ has the same sign with $p_j - p_{j'}$, and the desirable result then follows. ■

Lemma 18 *Let Γ be a Poisson-binomial random variable with the success probability $(p_j)_{j=1,\cdots,d}$, then for any $\tau \geq 1$, we have*

$$\Big(\sum_{j=1}^{d} p_j + \tau\Big)^{-1} \leq \mathbb{E}\Big(\frac{1}{\Gamma + \tau}\Big) \leq \Big(\frac{d+1}{d}\sum_{j=1}^{d} p_j + \tau - 1\Big)^{-1}.$$

Proof According to the corollary in Chao and Strawderman (1972), we have

$$\mathbb{E}\big(\frac{1}{\Gamma+\tau}\big) = \int_0^1 t^{\tau-1} P_\Gamma(t) dt = \int_0^1 t^{\tau-1} \big(\prod_{j=1}^d (1-p_j+p_j t) \big) dt \leq \int_0^1 t^{\tau-1} (1-\bar{p}+\bar{p}t)^d dt$$
$$= \mathbb{E}\big(\frac{1}{\Lambda+\tau}\big) \leq \big(\frac{d+1}{d} \sum_{j=1}^d p_j + \tau - 1\big)^{-1},$$

where $\bar{p} = d^{-1}\sum_{j=1}^d p_j$, the first inequality follows from the inequality of arithmetic and geometric means, the last equality follows from Section 3.1 of Chao and Strawderman (1972), and the last inequality follows from (25) in Wooff (1985). This completes the proof. ■

References

Miklós Ajtai, János Komlós, and Endre Szemerédi. An $O(n \log n)$ sorting network. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 1–9, 1983.

Abdulhakam AM Assidiq, Othman O Khalifa, Md Rafiqul Islam, and Sheroz Khan. Real time lane detection for autonomous vehicles. In *2008 International Conference on Computer and Communication Engineering*, pages 82–88. IEEE, 2008.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

Han Bao and Masashi Sugiyama. Calibrated surrogate maximization of linear-fractional utility in binary classification. In *International Conference on Artificial Intelligence and Statistics*, pages 2337–2347. PMLR, 2020.

Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Felix Belzunce, Carolina Martinez Riquelme, and Julio Mulero. *An Introduction to Stochastic Orders*. Academic Press, 2015.

Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018.

Jeroen Bertels, David Robben, Dirk Vandermeulen, and Paul Suetens. Optimization with soft dice can lead to a volumetric bias. In *International MICCAI Brainlesion Workshop*, pages 89–97. Springer, 2019.

Noah Bice, Neil Kirby, Ruiqi Li, Dan Nguyen, Tyler Bahr, Christopher Kabat, Pamela Myers, Niko Papanikolaou, and Mohamad Fakhreddine. A sensitivity analysis of probability maps in deep-learning-based anatomical segmentation. *Journal of Applied Clinical Medical Physics*, 22(8): 105–119, 2021.

Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier Transform and Its Applications*, volume 31999. McGraw-hill New York, 1986.

Min-Te Chao and WE Strawderman. Negative moments of positive random variables. *Journal of the American Statistical Association*, 67(338):429–431, 1972.

Nontawat Charoenphakdee, Jayakorn Vongkulbhisal, Nuttapong Chairatanakul, and Masashi Sugiyama. On focal loss for class-posterior probability estimation: A theoretical perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5202–5211, 2021.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

David F Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

Felipe Cucker and Ding Xuan Zhou. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.

Claudia D'Ambrosio, Silvano Martello, and Michele Monaci. Lower and upper bounds for the non-linear generalized assignment problem. *Computers & Operations Research*, 120:104933, 2020.

Jack Edmonds and Richard M Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

Evarist Giné and Vladimir Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *Annals of Probability*, 34(3):1143–1216, 2006.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE, 2011.

Yili Hong. On computing the distribution function for the poisson binomial distribution. *Computational Statistics & Data Analysis*, 59:41–51, 2013.

Shruti Jadon. Semsegloss: A python package of loss functions for semantic segmentation. *Software Impacts*, 9:100078, 2021.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

Yi Lin. A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68 (1):73–82, 2004.

Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pages 225–239. Springer, 2014.

Sheng Liu, Aakash Kaku, Weicheng Zhu, Matan Leibovich, Sreyas Mohan, Boyang Yu, Haoxiang Huang, Laure Zanna, Narges Razavian, Jonathan Niles-Weed, et al. Deep probability estimation. *arXiv preprint arXiv:2111.10734*, 2021.

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.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.

Katta G Murty and Santosh N Kabadi. Some np-complete problems in quadratic and nonlinear programming. Technical report, 1985.

Kritsana Neammanee. A refinement of normal approximation to poisson binomial. *International Journal of Mathematics and Mathematical Sciences*, 2005(5):717–728, 2005.

John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for? *Queue*, 6(2):40–53, 2008.

Marcus Nordström, Han Bao, Fredrik Löfman, Henrik Hult, Atsuto Maki, and Masashi Sugiyama. Calibrated surrogate maximization of Dice. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*, pages 269–278. Springer, 2020.

Yassine Ouali. Github repository: pytorch-segmentation. `https://github.com/yassouali/pytorch-segmentation`, 2022.

David Pollard. *Convergence of Stochastic Processes*. Springer Science & Business Media, 1984.

Teodora Popordanoska, Jeroen Bertels, Dirk Vandermeulen, Frederik Maes, and Matthew B Blaschko. On the relationship between calibrated predictors and unbiased volume estimation.

In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*, pages 678–688. Springer, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International Workshop on Machine Learning in Medical Imaging*, pages 379–387. Springer, 2017.

Xiaotong Shen. On methods of sieves and penalization. *Annals of Statistics*, 25(6):2555–2591, 1997.

Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5734–5743, 2017.

Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer, 2017.

Pejman Tahmasebi, Ardeshir Hezarkhani, and Muhammad Sahimi. Multiple-point geostatistical modeling based on the cross-correlation functions. *Computational Geosciences*, 16(3):779–797, 2012.

Ambuj Tewari and Peter L Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(5), 2007.

Nobuaki Tomizawa. On some techniques useful for solution of transportation network problems. *Networks*, 1(2):173–194, 1971.

Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1559–1572, 2018.

David A Wooff. Bounds on reciprocal moments with applications and developments in stein estimation and post-stratification. *Journal of the Royal Statistical Society: Series B (Methodological)*, 47(2):362–371, 1985.

Yang Xin, Lingshuang Kong, Zhi Liu, Chunhua Wang, Hongliang Zhu, Mingcheng Gao, Chensu Zhao, and Xiaoke Xu. Multimodal feature-level fusion for biometrics identification system on iomt platform. *IEEE Access*, 6:21418–21426, 2018.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–85, 2004.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.