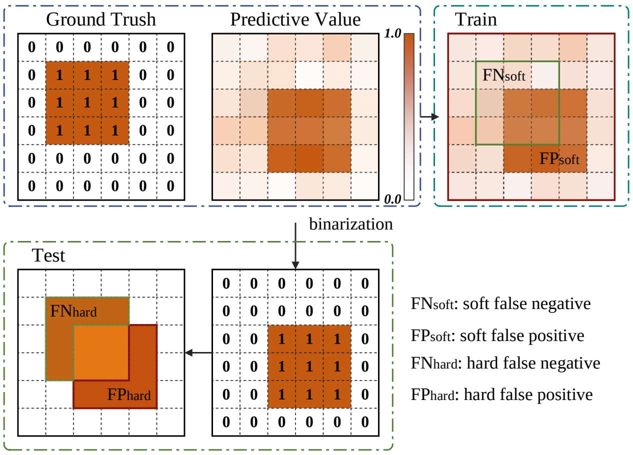

Figure 1: Illustration of FP and FN calculation bias during training and testing.

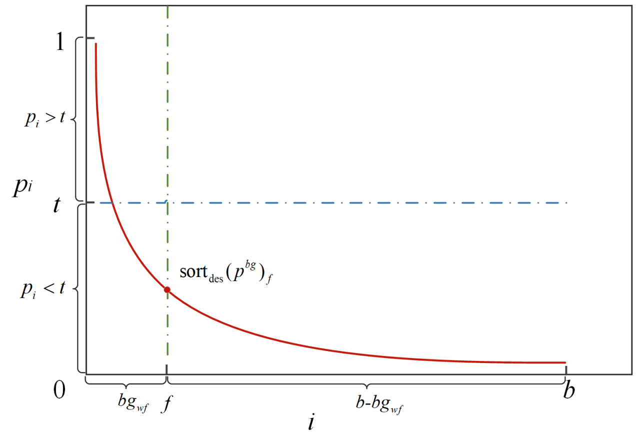

Figure 2: Sort the predicted values for the background region in descending order.

1 Appendix

FP and FN Calculation Bias Between Training and Testing Scenarios

As shown in Fig. 1, in the following discussion, both the predicted values and the ground truth values refer to the foreground. For convenience, the following discussion is conducted in the context of binary medical image 2D segmentation.

During testing, researchers typically binarize the predicted probability values into 0 and 1 based on a threshold t (usually $t = 0.5$) to classify pixels, and then calculate FP and FN. We refer to this method as hard calculation, and the FP and FN are called hard FP and hard FN. The formulas are as follows:

$$\mathrm{FP}_{\mathrm{hard}} = \sum_{i \in bg} \mathbb{I}\{p_i > t\}, \tag{1}$$

$$\mathrm{FN}_{\mathrm{hard}} = \sum_{i \in fg} \mathbb{I}\{(1 - p_i) \geq t\}, \tag{2}$$

where $i \in bg$ indicates that pixel i belongs to the background class, and $i \in fg$ indicates that pixel i belongs to the foreground class. $p_i \in [0, 1]$ represents the predictive value of the foreground by the model.

However, the network calculates FP and FN during training based on the probability values. We refer to this calculation method as soft calculation. The FP and FN are called soft FP and soft FN. The formulas are: $\mathrm{FP}_{\mathrm{soft}} = \sum_{i \in bg} p_i$, $\mathrm{FN}_{\mathrm{soft}} = \sum_{i \in fg}(1 - p_i)$. Specifically, $\mathrm{FP}_{\mathrm{soft}}$ and $\mathrm{FN}_{\mathrm{soft}}$ can be decomposed into two terms based on the threshold t:

$$\mathrm{FP}_{\mathrm{soft}} = \sum_{i \in bg} p_i \mathbb{I}\{p_i > t\} + \sum_{i \in bg} p_i \mathbb{I}\{p_i \leq t\}, \tag{3}$$

$$\begin{aligned} \mathrm{FN}_{\mathrm{soft}} = & \sum_{i \in fg} (1 - p_i) \mathbb{I}\{(1 - p_i) \geq t\} \\ & + \sum_{i \in fg} (1 - p_i) \mathbb{I}\{(1 - p_i) < t\}. \end{aligned} \tag{4}$$

The soft computation method may yield unequal results when the hard computation method results in equal FP and FN.

As shown in Fig. 2, let f represent the number of foreground pixels, and b represent the number of background pixels. Let $e_{bg}^{\bar{>}t}$ denote the average error between the predicted values and the ground truth for background pixels with predicted values greater than t:

$$e_{bg}^{\bar{>}t} = \frac{\sum_{i \in bg} p_i \mathbb{I}\{p_i > t\}}{\sum_{i \in bg} \mathbb{I}\{p_i > t\}} = \frac{\sum_{i \in bg} p_i \mathbb{I}\{p_i > t\}}{\mathrm{FP}_{\mathrm{hard}}}, \tag{5}$$

similarly:

$$e_{bg}^{\bar{\leq}t} = \frac{\sum_{i \in bg} p_i \mathbb{I}\{p_i \leq t\}}{\sum_{i \in bg} \mathbb{I}\{p_i \leq t\}} = \frac{\sum_{i \in bg} p_i \mathbb{I}\{p_i \leq t\}}{b - \mathrm{FP}_{\mathrm{hard}}}, \tag{6}$$

$$\begin{aligned} e_{fg}^{\bar{\geq}t} &= \frac{\sum_{i \in fg}(1 - p_i)\mathbb{I}\{(1 - p_i) \geq t\}}{\sum_{i \in fg} \mathbb{I}\{(1 - p_i) \geq t\}} \\ &= \frac{\sum_{i \in fg}(1 - p_i)\mathbb{I}\{(1 - p_i) \geq t\}}{\mathrm{FN}_{\mathrm{hard}}}, \end{aligned} \tag{7}$$

$$\begin{aligned} e_{fg}^{\bar{>}t} &= \frac{\sum_{i \in fg}(1 - p_i)\mathbb{I}\{(1 - p_i) < t\}}{\sum_{i \in fg} \mathbb{I}\{(1 - p_i) < t\}} \\ &= \frac{\sum_{i \in fg}(1 - p_i)\mathbb{I}\{(1 - p_i) < t\}}{f - \mathrm{FN}_{\mathrm{hard}}}. \end{aligned} \tag{8}$$

Substitute Eq. (5), Eq. (6), Eq. (7), and Eq. (8) into Eq. (3) and Eq. (4), resulting in:

$$\mathrm{FP}_{\mathrm{soft}} = \underbrace{e_{bg}^{\bar{>}t}\mathrm{FP}_{\mathrm{hard}}}_{\text{main}} + \underbrace{e_{bg}^{\bar{\leq}t}(b - \mathrm{FP}_{\mathrm{hard}})}_{\text{bias}}, \tag{9}$$

$$\mathrm{FN}_{\mathrm{soft}} = \underbrace{e_{fg}^{\bar{\leq}t}\mathrm{FN}_{\mathrm{hard}}}_{\text{main}} + \underbrace{e_{fg}^{\bar{>}t}(f - \mathrm{FN}_{\mathrm{hard}})}_{\text{bias}}. \tag{10}$$

According to Eq. (9) and Eq. (10), the main term is unaffected by inter-class imbalance, while the imbalance only

Algorithm 1: Two-stage training strategy

Input: $\mathcal{P} = \{p_1, \ldots, p_n\}$: A batch of network outputs predictive value of the foreground, $\mathcal{Y} = \{y_1, \ldots, y_n\}$: Ground truth of foreground, τ : A boolean value indicating whether the network has converged

Output:$\mathcal{L}$: Loss value, τ

if $\tau = false$ **then**
  Calculate $\mathcal{L}$ according to Intra-CBL;
  $num_f = 0$
  $num_b = 0$
  for i in $1, 2, \ldots, n$ **do**
    if $\mathrm{sort}_{\mathrm{des}}(y_i p_i)_f \geq t$ **then**
      $num_f \leftarrow num_f + 1$
    end if
    if $\mathrm{sort}_{\mathrm{des}}((1 - y_i)(1 - p_i))_f \geq t$ **then**
      $num_b \leftarrow num_b + 1$
    end if
  end for
  if $(num_f \geq \frac{n}{2}) \wedge (num_b \geq \frac{n}{2})$ **then**
    $\tau = true$
  end if
else
  Calculate $\mathcal{L}$ according to Balance loss;
end if
return $\mathcal{L}$, τ

impacts the bias terms. This explains why determining the weights for many re-weighted loss functions relies more on practice and why their parameters are challenging to generalize across different tasks. For example, with WCE loss and Tversky loss, whether adjusting for class weighting or weighting $\mathrm{FP}_{\mathrm{soft}}$ and $\mathrm{FN}_{\mathrm{soft}}$, the approach addresses inter-class imbalance by adjusting the ratio of $\mathrm{FP}_{\mathrm{soft}}$ and $\mathrm{FN}_{\mathrm{soft}}$. However, as shown by Eq. (9) and Eq. (10), changing the ratio of $\mathrm{FP}_{\mathrm{soft}}$ and $\mathrm{FN}_{\mathrm{soft}}$ also alters the ratio of the main terms, making it difficult to adjust bias terms solely based on ratios. Therefore, we propose Inter-CBL, which achieves inter-class balance by reducing the loss weight of $b - f$ simple pixels within the majority class.

However, during the early stages of training, Inter-CBL also affects the main terms, which is why Inter-CBL is described in the paper as unsuitable for training from scratch. This is because Inter-CBL alters $\mathrm{FP}_{\mathrm{soft}}$ to the following form:

if $\mathrm{sort}_{\mathrm{des}}(p^{bg})_f < t$:

$$\mathrm{FP}_{\mathrm{soft}} = e_{bg}^{\bar{>}t}\mathrm{FP}_{\mathrm{hard}} + e_{bg_{wf}}^{\leq\bar{t}}(f - \mathrm{FP}_{\mathrm{hard}}) + \epsilon e_{bg-bg_{wf}}^{\leq t\,\bar{}}(b - f) \tag{11}$$

if $\mathrm{sort}_{\mathrm{des}}(p^{bg})_f \geq t$:

$$\mathrm{FP}_{\mathrm{soft}} = e_{bg_{wf}}^{>\bar{t}} f + \epsilon\left(e_{bg-bg_{wf}}^{>t\,\bar{}}(\mathrm{FP}_{\mathrm{hard}} - f) + e_{bg}^{\bar{\leq}t}(b - \mathrm{FP}_{\mathrm{hard}})\right) \tag{12}$$

where ϵ is a minimal value, and the meaning of $\mathrm{sort}_{\mathrm{des}}(p^{bg})_f$ is illustrated in Fig. 2. When $\mathrm{sort}_{\mathrm{des}}(p^{bg})_f < t$, the form of $\mathrm{FP}_{\mathrm{soft}}$ is consistent with $\mathrm{FN}_{\mathrm{soft}}$, unaffected by class imbalance. However, when $\mathrm{sort}_{\mathrm{des}}(p^{bg})_f \geq t$, the main term is altered. In the early stages of training, $\mathrm{sort}_{\mathrm{des}}(p^{bg})_f$ is usually greater than t, which is why we state that Inter-CBL is not suitable for training from scratch.

Therefore, the Balance loss uses a two-stage approach for training. Training is conducted after the network has converged to an appropriate level. The specific process is illustrated in Algorithm 1.[1]

[1]Here, t refers to the parameter from Eq. (13) in the main text, not the threshold t used for binarizing the predicted probability values into 0 and 1.