Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Loss odyssey in medical image segmentation

Jun Ma[a,*], Jianan Chen[b], Matthew Ng[b], Rui Huang[b], Yu Li[a], Chen Li[d], Xiaoping Yang[d], Anne L. Martel[b,c]

[a] *Department of Mathematics, Nanjing University of Science and Technology, Nanjing, China*
[b] *Department of Medical Biophysics, University of Toronto, Toronto, Canada*
[c] *Physical Sciences, Sunnybrook Research Institute, Toronto, Canada*
[d] *Department of Mathematics, Nanjing University, Nanjing, China*

ARTICLE INFO

Article history:
Received 26 July 2020
Revised 4 March 2021
Accepted 6 March 2021
Available online 19 March 2021

Keywords:
Segmentation
Loss function
Convolutional neural networks
Benchmark

ABSTRACT

The loss function is an important component in deep learning-based segmentation methods. Over the past five years, many loss functions have been proposed for various segmentation tasks. However, a systematic study of the utility of these loss functions is missing. In this paper, we present a comprehensive review of segmentation loss functions in an organized manner. We also conduct the first large-scale analysis of 20 general loss functions on four typical 3D segmentation tasks involving six public datasets from 10+ medical centers. The results show that none of the losses can consistently achieve the best performance on the four segmentation tasks, but compound loss functions (e.g. Dice with TopK loss, focal loss, Hausdorff distance loss, and boundary loss) are the most robust losses. Our code and segmentation results are publicly available and can serve as a loss function benchmark. We hope this work will also provide insights on new loss function development for the community.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Loss functions, which aim to measure the dissimilarity between the ground truth and the predicted segmentation, play an essential role in modern convolutional neural networks (CNNs)-based segmentation methods. In CNN-based segmentation approaches, a simple recipe can be summarized as follows:

- Create a dataset; In general, well-labelled image-ground truth pairs are generated by manual or semi-automatic segmentation. Creating a large dataset is laborious and time-consuming, but now there are some public datasets for benchmarking different segmentation methods thanks to several segmentation challenge organizers (Menze et al., 2015; Bilic et al., 2019; Simpson et al., 2019; Zhuang et al., 2019).
- Build a neural network architecture; U-Net (Ronneberger et al., 2015) is one of the most popular CNNs for medical image segmentation which was introduced in 2015. During the past five years, a number of U-Net variants were proposed for various segmentation tasks, such as Res-UNet (Xiao et al., 2018), Dense U-Net (Guan et al., 2019), Hybrid Dense U-Net (Li et al., 2018b) MultiResU-Net (Ibtehaz and Rahman, 2020), Attention U-Net (Schlemper et al., 2019) and so on.
- Design a loss function; During the training phase, the loss function is used to guide the network to learn meaningful predictions that are close to the ground truth in terms of segmentation metrics, such as Dice similarity coefficient (DSC). Moreover, the loss function also dictates how the network is supposed to trade off mistakes (for example false positives, false negatives).
- Define the optimizer. The loss function is minimized by the employed optimizer. Stochastic gradient descent method and its variants (e.g., Adam (Kingma and Ba, 2015)) are the most popular choices.

This paper mainly focuses on the loss functions, which are important parts in CNN-based segmentation methods. Recently, cross entropy and Dice loss have become the most commonly used loss functions in medical image segmentation tasks (Milletari et al., 2016). For example, in the proceedings of MICCAI 2018, 47 out of 77 CNN-based segmentation papers (Bertels et al., 2019) chose cross entropy loss as their target loss. Dice loss has been widely used in many top solutions of medical image segmentation challenges (Bernard et al., 2018; Bakas et al., 2018).

In addition to cross entropy and Dice loss, more than 10 loss functions have been proposed for segmentation CNNs. Most of them are designed to address the class imbalance problem, which is one of the main challenges when the object to be segmented is small relative to the size of the image volume, for example the segmentation of tumors or small organs. Moreover, these loss func-

[*] Corresponding author.
E-mail address: junma@njust.edu.cn (J. Ma).

https://doi.org/10.1016/j.media.2021.102035
1361-8415/© 2021 Elsevier B.V. All rights reserved.

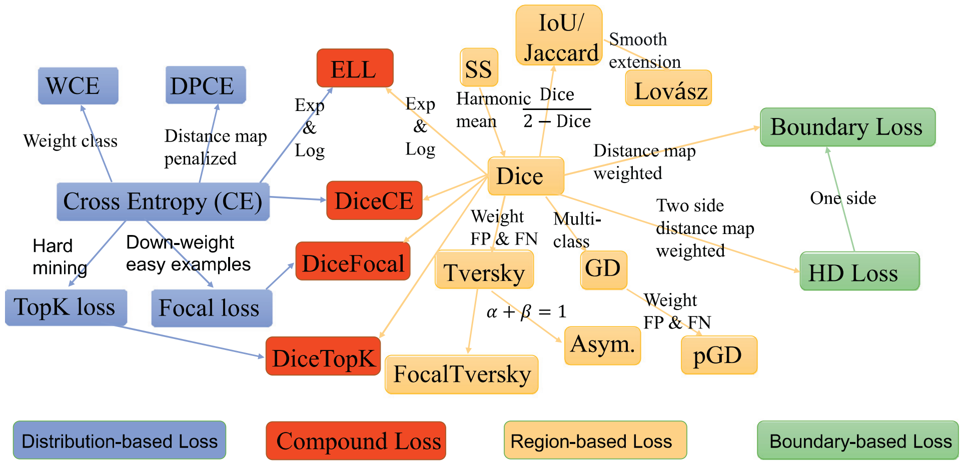


Fig. 1. Overview of 20 loss functions for medical image segmentation.

tions are usually agnostic to network architectures, and can be used for any segmentation tasks in a plug-and-play way.

We have witnessed the popularity of medical image segmentation challenges during the past years. These challenges serve as public benchmarks to evaluate and compare different segmentation methods proposed by researchers around the world. However, to the best of our knowledge, there is no comprehensive comparison and evaluation of these loss functions. Most existing loss functions are proposed and evaluated with different network structures as well as on different datasets. Moreover, all the studies only compared their proposed loss function with a limited number of alternative loss functions.

Given the diversity of the segmentation tasks and loss functions, it becomes increasingly difficult to identify which loss function can achieve the best performance with the same CNN structure. In this paper, we focus on the plug-and-play loss functions that can be used in any segmentation tasks, and aim to answer the following question:

Which loss function should we choose for medical image segmentation tasks?

Specifically, we first review 20 different loss functions systematically and then evaluate them on four well-known segmentation tasks based on six public datasets. Finally, we rank them according to their evaluation results.

The main contributions of this work are summarized as follows:

- We present the first comprehensive review and comparison of the existing plug-and-play loss functions in an organized manner.
- We also conduct a large set of experiments for 20 loss functions on four segmentation tasks with six public datasets from 10+ medical centers, and highlight the most robust loss functions.
- We build a loss function benchmark library by making the code, dataset splits, and segmentation results are publicly available at https://github.com/JunMa11/SegLoss, which could greatly advance new loss function development in the community.

The paper is organized as follows. Section 2 presents a taxonomy for 20 loss functions, and Section 3 introduces how do we built a fair experimental setting to evaluate these loss functions based on four popular segmentation tasks. We summarize and compare the experimental results of different loss function in Section 4. Section 5 discusses the ranking stability, the loss function relationship, and the limitations. Section 6 gives the final conclusion about practical loss function recommendations.

2. Loss function taxonomy

We classify loss functions into four categories based on how they are derived, namely, the mismatch in distribution, region, boundary or some combination of these. Moreover, we explore the relationships between these loss functions. Fig. 1 shows the four categories and the connections between loss functions.

Let I be an image on a domain $\Omega \subset \mathbf{R}^2$ or $\mathbf{R}^3$, and S, G denote the corresponding segmentation result and ground truth, respectively. s_i, g_i denote the predicted segmentation and ground truth of voxel i, respectively. N is the number of voxels in the image I, and C is the number of classes. In the following six subsections, we present the key ideas, formulations, and relationships between each other.

2.1. Distribution-based Loss

Distribution-based loss functions aim to minimize dissimilarity between two distributions. The most fundamental function in this category is cross entropy; all other functions are derived from cross entropy.

2.1.1. Cross entropy

Cross entropy (CE) is derived from Kullback-Leibler (KL) divergence, a measure of dissimilarity between two distributions P and Q, which is defined by

$$
\begin{aligned}
D_{KL}(P|Q) &= \sum_i p_i \log \frac{p_i}{q_i} \\
&= -\sum_i p_i \log q_i + \sum_i p_i \log p_i \\
&= H(P, Q) - H(P),
\end{aligned}
$$

where $H(P, Q) = -\sum_i p_i \log q_i$ is the cross entropy between the distribution P and Q, and $H(P) = -\sum_i p_i \log p_i$ is the entropy of the distribution P. For common machine learning tasks, the data distribution P is assumed to be given by the training set. Thus, minimizing KL divergence between the ground truth distribution P and predicted distribution Q is equivalent to minimizing the cross entropy $H(P, Q)$. For a CNN-based segmentation task, the cross entropy loss is defined by

$$L_{CE} = -\frac{1}{N}\sum_{c=1}^{C}\sum_{i=1}^{N} g_i^c \log s_i^c, \tag{1}$$

where g_i^c is the ground truth binary indicator of class label c of voxel i, and s_i^c is the corresponding predicted segmentation probability.

Weighted cross entropy (WCE) is a commonly used extension of CE, which is defined by

$$L_{WCE} = -\frac{1}{N}\sum_{c=1}^{C}\sum_{i=1}^{N} w_c g_i^c \log s_i^c, \tag{2}$$

where w_c is the weight for each class. Usually, w_c is inversely proportional to the class frequencies to penalize majority classes. In the following experiments, we set the class weight w_c as the reciprocal of the class frequency in the training set. Moreover, an alternative way to use weighted cross entropy is to assign a weight for each pixel that is computed based on the ground truth (Ronneberger et al., 2015).

2.1.2. TopK loss

TopK loss is also a variant of cross entropy, and aims to force networks to focus on hard samples during training. There are two implementations. One retains only the voxels with probability values lower than a given threshold (Wu et al., 2016), which is defined by

$$L_{TopK-thr} = -\frac{\sum\limits_{c=1}^{C}\sum\limits_{i=1}^{N} 1\{g_i = c \textit{ and } s_i^c < t\} \log s_i^c}{\sum\limits_{c=1}^{C}\sum\limits_{i=1}^{N} 1\{g_i = c \textit{ and } s_i^c < t\}}, \tag{3}$$

where $t \in (0, 1]$ is a threshold and $1\{\dots\}$ is the binary indicator function. In other words, “easy” pixels, i.e., the pixels with probability above t, are dropped as they have been easily classified by the model. The other retains the $k\%$ worst pixels for loss, irrespective of their loss/probability values, which is defined by

$$L_{TopK} = -\frac{1}{N}\sum_{c=1}^{C}\sum_{i\in\mathbf{K}} g_i^c \log s_i^c, \tag{4}$$

where $\mathbf{K}$ is the set of the $k\%$ worst pixels. In Section 2.1.2, we compare the two different implementations with different threshold t and and percentage k on four segmentation tasks. The percentage variant of TopK loss with $k = 10\%$ is our default setting.

2.1.3. Focal loss

Focal loss (Goyal and Kaiming, 2018) adapts the standard cross entropy to focus on hard examples by reducing the loss assigned to well-classified examples, which can also deal with foreground-background class imbalance. It is defined by

$$L_{Focal} = -\frac{1}{N}\sum_{c}^{C}\sum_{i=1}^{N}(1 - s_i^c)^\gamma g_i^c log s_i^c, \tag{5}$$

where $\gamma = 2$ obtains the best performance in the original paper.

2.1.4. Distance map penalized cross entropy loss (DPCE)

DPCE loss (Caliva et al., 2019) weights cross entropy by distance maps that are derived from ground truth masks. It aims to guide the network's focus towards hard-to-segment boundary regions, and is defined by

$$L_{DPCE} = -\frac{1}{N}\sum_{c=1}^{C}(1 + D^c) \circ \sum_{i=1}^{N} g_i^c log s_i^c, \tag{6}$$

where D^c is the distance penalty term of class c, and $\circ$ is the Hadamard product. Specifically, D^c is generated by taking the inverse of the distance transform of ground truth[1]. In this way, the pixels on the boundary can be assigned with larger weights.

2.2. Region-based Loss

Region-based loss functions aim to minimize the mismatch or maximize the overlap regions between ground truth G and predicted segmentation S. The popular Dice loss is the best known representative of this class of loss functions.

2.2.1. Sensitivity-specificity loss

Sensitivity-specificity loss (Brosch et al., 2015) addresses the class imbalance problem by weighting specificity higher; it is defined by

$$\begin{aligned} L_{SS} = w\tfrac{\sum_{c=1}^{C}\sum_{i=1}^{N}(g_i^c - s_i^c)^2 g_i^c}{\sum_{c=1}^{C}\sum_{i=1}^{N} g_i^c + \epsilon} \\ + (1 - w)\tfrac{\sum_{c=1}^{C}\sum_{i=1}^{N}(g_i^c - s_i^c)^2 (1 - g_i^c)}{\sum_{c=1}^{C}\sum_{i=1}^{N}(1 - g_i^c) + \epsilon}, \end{aligned} \tag{7}$$

where the parameter w controls the trade-off between sensitivity (the first term) and specificity (the second term).

2.2.2. Dice loss

Dice loss can directly optimize the Dice Similarity Coefficient (DSC) which is the most commonly used segmentation evaluation metric. Unlike weighted cross entropy, Dice loss does not require class re-weighting for imbalanced segmentation tasks. In general, there are two variants for Dice loss (Isensee et al., 2021), one employs squared terms in the denominator (Milletari et al., 2016), which is defined by

$$L_{Dice-square} = 1 - \frac{2\sum_{c=1}^{C}\sum_{i=1}^{N} g_i^c s_i^c}{\sum_{c=1}^{C}\sum_{i=1}^{N}(g_i^c)^2 + \sum_{c=1}^{C}\sum_{i=1}^{N}(s_i^c)^2}. \tag{8}$$

The other does not use the squared terms in the denominator (Drozdzal et al., 2016), which is defined by

$$L_{Dice} = 1 - \frac{2\sum_{c=1}^{C}\sum_{i=1}^{N} g_i^c s_i^c}{\sum_{c=1}^{C}\sum_{i=1}^{N} g_i^c + \sum_{c=1}^{C}\sum_{i=1}^{N} s_i^c}. \tag{9}$$

In the following experiments, we use the no-squared version as the default setting. We also quantitatively compare the two variants on four datasets in Section 4.3.

2.2.3. IoU (Jaccard) loss

Intersection over Union (IoU) loss (Rahman and Wang, 2016), similar to Dice loss, is also used to directly optimize the object category segmentation metric. It is defined by

$$L_{IoU} = 1 - \frac{\sum_{c=1}^{C}\sum_{i=1}^{N} g_i^c s_i^c}{\sum_{c=1}^{C}\sum_{i=1}^{N}(g_i^c + s_i^c - g_i^c s_i^c)}. \tag{10}$$

2.2.4. Lovász loss

Similar to IoU loss, Lovasz loss is also a surrogate to directly optimize the Jaccard index, but it uses different substitution strategy. In particular, IoU loss (also known as soft Jaccard) simply replaces the segmentation in Jaccard index with softmax probabilities, while Lovász loss (Berman et al., 2018) uses a piecewise linear convex surrogate to the IoU loss based on the Lovász extension of submodular set functions[2]. Specifically, a vector of pixel errors $m(c)$ for class $c \in C$ is defined by

$$m_i(c) = \begin{cases} 1 - s_i^c, & if \quad c = g_i \\ s_i^c, & otherwise \end{cases} \tag{11}$$

Then, the vector of errors $m(c) \in [0, 1]^p$ is used to construct the loss surrogate

$$L_{lovasz} = \overline{\triangle J_c}(m(c)) \tag{12}$$

[1] scipy.ndimage.morphology.distance_transform_edt

[2] This is because IoU loss is submodular.

where $\overline{\triangle J_c}$ is the lovász extension of the (discrete formulation) Jaccard function $\triangle : \{0,1\}^N \rightarrow \frac{|m_c|}{|\{g_i=c\} \cup m_c|}$ which is defined by

$$\overline{\triangle} : m \in R^p \mapsto \sum_{i=1}^{N} m_i d_i(m) \tag{13}$$

with $d_i(m) = \triangle(\{\pi_1, \ldots, \pi_i\}) - \triangle(\{\pi_1, \ldots, \pi_{i-1}\})$, π being a permutation ordering the components of m in decreasing order, i.e. $x_{\pi_1} \geq x_{\pi_2} \cdots \geq x_{\pi_N}$.

2.2.5. Tversky loss

To achieve a better trade-off between precision and recall, Tversky loss (Salehi et al., 2019) adapts the Dice loss to emphasize false negatives; it is defined by

$$\begin{aligned} L_{Tversky} &= 1 - T(\alpha, \beta) \\ &= 1 - (\textstyle\sum_c^C \sum_{i=1}^N g_i^c s_i^c)/(\sum_c^C \sum_{i=1}^N g_i^c s_i^c \\ &\quad + \alpha \textstyle\sum_c^C \sum_{i=1}^N (1 - g_i^c) s_i^c + \beta \sum_c^C \sum_{i=1}^N g_i^c (1 - s_i^c)) \end{aligned}, \tag{14}$$

where α and β are hyper-parameters that control the tradeoff between false negatives and false positives.

2.2.6. Generalized Dice loss

Generalized Dice loss (Sudre et al., 2017) is the multi-class extension of Dice loss where the weight of each class is inversely proportional to the label frequencies. It can be expressed by

$$L_{GD} = 1 - 2\frac{\sum_{c=1}^C w_c \sum_{i=1}^N g_i^c s_i^c}{\sum_{c=1}^C w_c \sum_{i=1}^N (g_i^c + s_i^c)} \tag{15}$$

where $w_c = \frac{1}{(\sum_{i=1}^N g_i^c)^2}$ is used to provide invariance to different label set properties.

2.2.7. Focal Tversky loss

Focal Tversky loss (Abraham and Khan, 2019) applies the concept of focal loss to focus on hard cases with low probabilities; it is defined by

$$L_{FTL} = (L_{Tversky})^{\frac{1}{\gamma}}, \tag{16}$$

where γ varies in the range of $[1,3]$.

2.2.8. Asymmetric similarity loss

Dice loss can be regarded as the harmonic mean of precision and recall and it weights false positives (FPs) and false negatives (FNs) equally. The motivation of asymmetric similarity loss (Hashemi et al., 2019) is to make a better adjustment of the weights of FPs and FNs (and achieve a better balance between precision and recall) by introducing a weighting parameter β, which is defined by

$$\begin{aligned} L_{Asym} = &(\textstyle\sum_{c=1}^C \sum_{i=1}^N g_i^c s_i^c)/(\sum_{c=1}^C \sum_{i=1}^N g_i^c s_i^c \\ &+ \tfrac{\beta^2}{1+\beta^2} \textstyle\sum_{c=1}^C \sum_{i=1}^N g_i^c (1 - s_i^c) + \tfrac{1}{1+\beta^2} \sum_{c=1}^C \sum_{i=1}^N (1 - g_i^c) s_i^c), \end{aligned} \tag{17}$$

The recommended β is 1.5 in the original paper(Hashemi et al., 2019). It should be noted that asymmetric similarity loss is also a special case of Tversky loss when $\alpha + \beta = 1$ in Eq. (14).

2.2.9. Penalty loss

To penalize the false negatives and the false positives in generalized Dice (GD), (Su et al., 2019) proposed the penalty loss $L_{pGD} = 1 - pGD$ where the pGD is defined by

$$\begin{aligned} pGD &= 2(\textstyle\sum_{c=1}^C w_c \sum_{i=1}^N g_i^c s_i^c)/\Big(\sum_{c=1}^C w_c \sum_{i=1}^N (g_i^c + s_i^c) \\ &\quad + k \textstyle\sum_{c=1}^C w_c \sum_{i=1}^N (1 - g_i^c) s_i^c + k \sum_{c=1}^C w_c \sum_{i=1}^N g_i^c (1 - s_i^c) \Big) \\ &= \frac{2 \sum_{c=1}^C w_c \sum_{i=1}^N g_i^c s_i^c}{\sum_{c=1}^C w_c \sum_{i=1}^N (g_i^c + s_i^c) + k \sum_{c=1}^C w_c \sum_{i=1}^N (s_i^c - 2 s_i^c g_i^c + g_i^c)} \\ &= \frac{\frac{2 \sum_{c=1}^C w_c \sum_{i=1}^N g_i^c s_i^c}{\sum_{c=1}^C w_c \sum_{i=1}^N (g_i^c + s_i^c)}}{\frac{\sum_{c=1}^C w_c \sum_{i=1}^N (g_i^c + s_i^c)}{\sum_{c=1}^C w_c \sum_{i=1}^N (g_i^c + s_i^c)} + \frac{k \sum_{c=1}^C w_c \sum_{i=1}^N (s_i^c - 2 s_i^c g_i^c + g_i^c)}{\sum_{c=1}^C w_c \sum_{i=1}^N (g_i^c + s_i^c)}} \\ &= \frac{GD}{1 + k(1 - GD)}, \end{aligned} \tag{18}$$

where $\sum_{c=1}^C w_c \sum_{i=1}^N (1 - g_i^c) s_i^c$ and $\sum_{c=1}^C w_c \sum_{i=1}^N g_i^c (1 - s_i^c)$ are the summations of false negatives and false positives, respectively, $w_c = \frac{1}{(\sum_{i=1}^N g_i^c)^2}$ is the weight of different classes, k is a non-negative penalty coefficient. When $k = 0$, pGD is equivalent to generalized Dice. When $k > 0$, pGD gives additional weights to false positives and false negatives. In the original paper (Su et al., 2019), $k = 2.5$ corresponds the best performance.

2.3. Boundary-based Loss

Boundary-based loss, a relatively new type of loss function, aims to minimize the distance between ground truth and predicted segmentation.

2.3.1. Boundary (BD) loss

There are two different frameworks to compute the distance between two boundaries. One is the differential framework that computes the motion of each point on the boundary curve as a velocity along the normal to the curve. The other is the integral framework that approximates the distance by computing the integrals over the interface between mismatch regions of the two boundaries. The differential framework cannot be used directly as a loss for the network softmax output because it is non-differentiable. To compute the distance $Dist(\partial G, \partial S)$ between two boundaries in a differential way, boundary loss (Kervadec et al., 2019; 2021) uses the integral framework to formulate the boundary loss, which can avoid local differential computations involving boundary curve points. Formally, we have

$$\begin{aligned} Dist(\partial G, \partial S) &= \textstyle\int_{\partial G} ||q_{\partial S}(p) - p||^2 dp \\ &\approx 2 \textstyle\int_{\Delta S} D_G(p) dp \\ &= 2(\textstyle\int_{\Omega} \phi_G(p) s(p) dp - \int_{\Omega} \phi_G(p) g(p) dp)) \end{aligned}, \tag{19}$$

where $\Delta M = (S/G) \cup (G/S)$ is the inconsistency part between ground truth and segmentation, $D_G(p)$ is the distance map of ground truth, $s(p)$ and $g(p)$ are binary indicator functions. ϕ_G is the level set representation of boundary: $\phi_G = -D_G(q)$ if $q \in G$, and $\phi_G = D_G(q)$ otherwise. In this way, the integral is over the boundary rather than the unbalanced regions, which can mitigate the difficulties of highly unbalanced segmentation. Furthermore, $s(p)$ in Eq. (19) is replaced by the softmax probability outputs $s_\theta(p)$ of the network to form a trainable function. The last term is omitted as it is independent to the network parameters. Finally, we obtain the boundary loss function as follows[3]:

$$L_{BD} = \sum_{\Omega} \phi_G(p) s_\theta(p). \tag{20}$$

2.3.2. Hausdorff Distance (HD) loss

Hausdorff distance, a boundary-based metric, is widely used for evaluating segmentation methods. However, directly minimizing HD during training is intractable and could lead to unstable training. To address this problem, Karimi et al. (2020) show that HD can be approximated by the distance transforms of ground truth

[3] For notation harmonization, we use sum to replace the integral.

and predicted segmentation. The network can be trained with following HD loss function for reducing HD:

$$L_{HD} = \frac{1}{N}\sum_{c=1}^{C}\sum_{i=1}^{N}[(s_i^c - g_i^c)\circ(d_{G_i^c}^2 + d_{S_i^c}^2)], \tag{21}$$

where d_G and d_S are distance transform maps of ground truth and segmentation, respectively. Specifically, the distance transform computes the shortest distance between each pixel and the object boundary[4].

It should be noted that both boundary loss (Kervadec et al., 2019) and Hausdorff Distance loss (Karimi and Salcudean, 2020) should be coupled with region-based loss (e.g., Dice loss) when being used for training neural networks, so as to reduce the training instability.

2.4. Compound Loss

Compound loss is the (weighted) combination between the above loss functions.

2.4.1. Combo loss

Combo loss (Isensee et al., 2019; Taghanaki et al., 2019) is the sum between cross entropy and Dice loss; it is defined by

$$L_{DiceCE} = L_{CE} + L_{Dice}. \tag{22}$$

2.4.2. Exponential Logarithmic loss (ELL)

To address the issues of highly unbalanced object sizes, Wong et al. (2018) proposed to make exponential and logarithmic transformations to both Dice loss and cross entropy loss. In this way, the network can be forced to intrinsically focus more on less accurately predicted structures. The Exponential Logarithmic loss is defined by

$$\begin{aligned} L_{ELL} = {} & w_{Dice}E[(-log(Dice_c))^{\gamma Dice}] \\ & + w_{CE}E[w_c(-log(s_i^c))^{\gamma CE}], \end{aligned} \tag{23}$$

where $Dice_c = \frac{2\sum_{i=1}^{N} g_i^c s_i^c + \epsilon}{\sum_{i=1}^{N}(g_i^c + s_i^c) + \epsilon}$.

2.4.3. Dice loss with focal loss

To alleviate the imbalanced organ segmentation problem and force the model to learn from poorly segmented voxels better, (Zhu et al., 2019) employ a hybrid loss consisting of both Dice loss and focal loss; it is defined by

$$L_{DiceFocal} = L_{Dice} + L_{Focal}, \tag{24}$$

where L_{Dice} and L_{Focal} are defined in Eqs. (9) and (5), respectively.

2.4.4. Dice loss with TopK loss

Recently, Dice loss is used in conjunction with a TopK loss for automated volumetric assessment of multiple sclerosis (Brugnara et al., 2020); it is defined by

$$L_{DiceTopK} = L_{Dice} + L_{TopK}. \tag{25}$$

2.5. Beyond plug-and-play loss functions: tailored loss

In addition to the above plug-and-play loss functions that can be used in any segmentation tasks, there are also some tailored loss functions for special segmentation tasks. For example, generalized Wasserstein Dice loss (Fidon et al., 2017) is used for improving multi-class segmentation by exploring label relationships. Some shape priors and constraints are also incorporated into loss functions to enforce the segmentation results with desired topological structures. For example, to improve the network's region labelling consistency, (Ganaye et al., 2019) proposed an adjacency-graph based auxiliary training loss that can penalize outputs containing regions with anatomically-incorrect adjacency relationships. (Hu et al., 2019) designed a continuous-valued loss function that can enforce the segmentation to have the same Betti number as the ground truth. This paper aims to evaluate the plug-and-play loss functions, and a detailed explanation of these customized loss functions is beyond the main scope of this paper.

2.6. Connections among Dice loss, boundary loss and Hausdorff distance loss

Although boundary loss and Hausdorff distance loss aim to reduce the mismatch between boundaries during training, both of them are computed in a region-based way. This subsection aims to explore the connections among Dice loss, boundary loss and Hausdorff distance loss. We use $\Delta M = (S/G) \cup (G/S)$ to denote the mismatched area between ground truth and predicted segmentation. For ease of illustration, we focus on binary segmentation. The three loss functions can be reformulated as:

- Dice loss: $L_{Dice} = 1 - \frac{2|G\cap S|}{|G|+|S|} = \frac{|\Delta M|}{|G|+|S|}$;
- Boundary loss: $L_{BD} \approx \int_{\Delta M} D_G(p)dp$;
- Hausdorff distance loss: $L_{HD} \approx \frac{1}{|\Omega|}\sum_{\Omega} \Delta M \circ (D_G + D_S)$.

It can be found that all three loss functions could be used to minimize the mismatched region ΔM between the ground truth and the predicted segmentation. The key difference among them is the weights of ΔM. In particular, for Dice loss, the segmentation mismatch is weighted by the sum between the number of object pixels in the segmentation and the number of pixels in the ground truth. For boundary loss, the mismatched region ΔM is weighted by the distance transform map of ground truth. For the Hausdorff distance loss, the weights are based on not only the distance transform map of the ground truth but also the distance transform map of the segmentation.

3. Loss function evaluation

3.1. Tasks and datasets

We evaluate the loss functions on four different segmentation tasks involving balanced and imbalanced foreground-background, binary and multi-class segmentation. Table 1 presents an overview of the tasks and datasets. These datasets are from 10+ medical centers around the world, and we present the details as follows. The liver segmentation dataset has a mild label imbalance, while the other three datasets are highly imbalanced.

3.1.1. Liver segmentation dataset

The data is adapted from the liver tumor segmentation benchmark dataset that was collected from seven academic and clin-

Table 1
Overview of the tasks and datasets. The Num denotes the number of image cases in each set. BG:FG denotes the ratio between the number of voxels in background (BG) and foreground (FG). In the multi-organ dataset, the ratio ranges from 30 : 1 to 1243 : 1. The liver has a mild imbalance, while the gallbladder is a highly imbalanced problem.

Task	Num	BG:FG	Dataset
Liver	131	42:1	LiTS
Liver Tumor	434	576:1	LiTS and MSD
Pancreas	363	503:1	NIH and MSD
Multi-organ	90	30:1-1243:1	NIH and BTCV

[4] There is an out-of-the-box implementation of the distance transform in scipy: scipy.ndimage.morphology.distance transform_edt.

ical institutions around the world (Bilic et al., 2019). There are 131 contrast-enhanced abdominal CT scans in the training set. The inter-plane resolution is from 0.45mm to 6.0mm and intra-slice spacing is from 0.55mm to 1.0mm.

3.1.2. Liver tumor segmentation dataset

We collect liver tumor cases from both LiTS (Bilic et al., 2019) and Decathlon (Simpson et al., 2019) Task08_Hepatic Vessel dataset, and the total number of CT cases is 434. We still include the LiTS dataset in this experiment but the corresponding ground truth is liver tumor rather than the liver. Task08_Hepatic Vessel dataset consists of 303 training cases that are provided by Memorial Sloan Kettering Cancer Center (New York, NY, USA). All the cases in Task08_Hepatic Vessel are from the portal venous phase.

3.1.3. Pancreas segmentation dataset

We group pancreas cases from both National Institutes of Health (NIH) Clinical Center Pancreas dataset[5] and Decathlon Task07_Pancreas dataset (Simpson et al., 2019). The total number of cases is 363. Specifically, the NIH pancreas dataset consists of 82 abdominal contrast enhanced 3D CT scans (70 s after intravenous contrast injection in portal-venous) and the Decathlon Task07_Pancreas dataset includes 281 portal venous phase CT scans.

3.1.4. Multi-organ segmentation dataset

This dataset includes 90 multi-organ abdominal CT cases that are from NIH Pancreas-CT and Beyond the Cranial Vault (BTCV) segmentation challenge (Landman et al., 2015), and the corresponding ground truth[6] is released in conjunction with the Dense V-Networks (Gibson et al., 2018). The label comprises eight organs: spleen, left kidney, gallbladder, esophagus, liver, stomach, pancreas and duodenum.

3.2. Network architecture and training protocol

The goal of this work was to study the impact of different loss functions. We decide to use a typical 3D U-Net (Çiçek et al., 2016) and standard training procedure, which can reduce the impact of other factors that may confound the results. In particular, we use nnU-Net V1 as presented in (Isensee et al., 2019) as the network backbone because it has been shown to achieve state-of-the-art performance on more than 10 public segmentation datasets. During preprocessing, training, and testing, we use the default settings in nnU-Net except the learning rate. This is because that the initial learning rate is an important hyper parameter that can have a crucial impact on the final performance. For a fair comparison between different loss functions, we search the best learning rate for each loss in the set $\{3e-3, 3e-4, 3e-5\}$, and the best performing model is used during testing. We conduct all the experiments on the same type of GPU (TITAN V100).

For the sake of completeness, we also briefly introduce some important settings in nnU-Net.

- **Preprocessing:** We resample all cases to the median voxel spacing on each dataset, where 3rd-order spline interpolation is used for scans and nearest neighbor interpolation is used for the corresponding ground truth. In addition, the intensity of each case is normalized by cutting off to the $[0.5, 99.5]$ percentiles of the whole Hounsfield Units, followed by a Z-Score normalization based on the mean and standard deviation of the intensity values. After that, we randomly split each dataset into 80% for training and 20% for testing.
- **Data augmentation:** To avoid overfitting, data augmentation methods are applied on the fly during training, including randomly mirroring along all axes, random cropping, scaling, rotation and deformation, and gamma transformation. In addition, oversampling of foreground regions is employed to eliminate the label imbalance. Specifically, 33% of the samples in a mini-batch are guaranteed to contain at least one of the foreground classes.
- **Optimization:** All the losses are minimized by stochastic descent with the Adam optimizer (Kingma and Ba, 2015) (β_1=0.9, β_2=0.999) and a mini-batch size of 2. The training process takes up to 1000 epochs, where one epoch is defined as the iteration over 250 training batches. Whenever the moving average of the training loss does not improve for 50 epochs, the learning rate is reduced by multiplying it with 0.2 until a minimum learning rate of 1e-6 has been reached.

In addition, three loss functions require additional scheduling strategies during training, including Lovász loss, boundaries loss, and Hausdorff distance loss. Specifically, for the Lovász loss, the authors in (Berman et al., 2018) suggest optimizing with cross entropy first and then finetuning with the Lovász loss. The boundary loss (Kervadec et al., 2019) and Hausdorff distance loss (Karimi and Salcudean, 2020) should be combined with the Dice loss as follows:

$$L = \alpha L_{Dice} + \beta L_{(\cdot)},$$

where $\alpha, \beta > 0$ is a weight hyper-parameter, and $(\cdot)$ denotes boundary loss or Hausdorff distance loss. For the boundary loss, (Kervadec et al., 2019) suggested using the Dice loss to dominate initial training so as to stabilize the training process and quickly obtain a reasonable initial segmentation. Specifically, they set the weight $\beta = 1 - \alpha$ and $\alpha = 1$ initially, and decrease α by 0.01 after each epoch until it reaches the value of 0.01. For the Hausdorff distance loss, (Karimi and Salcudean, 2020) set α to be the ratio of the mean of the HD-based loss term to the mean of the DSC loss term, and $\beta = 1$. A recent empirical study (Ma et al., 2020) has shown that the implementation details can lead to a noticeable impact on the performance. In our experiments, we first train the network with the Dice loss, and then finetune[7] the BD loss and HD loss with the suggested scheduling strategies, because we found that this training trick can obtain robust training process and also give the best performance. All the other 17 loss function can be used in a plug-and-play way without any specific scheduling tricks during training.

3.3. Quantitative metrics

Two complementary metrics are used to evaluate the segmentation results. Dice coefficient, a region-based metric, is used to evaluate the region overlap. Surface dice (Nikolov et al., 2018), a boundary-based metric is used to evaluate how close the segmentation and ground truth surfaces are to each other at a specified tolerance.

3.3.1. Region-based metric

Dice similarity coefficient.

$$DSC(G, S) = \frac{2|G \cap S|}{|G| + |S|}$$

3.3.2. Boundary-based metric

Normalized Surface Distance (NSD) at tolerance τ.

$$NSD(G, S) = \frac{|\partial G \cap B_{\partial S}^{(\tau)}| + |\partial S \cap B_{\partial G}^{(\tau)}|}{|\partial G| + |\partial S|}$$

[5] https://wiki.cancerimagingarchive.net/display/Public/Pancreas-CT

[6] http://doi.org/10.5281/zenodo.1169361

[7] The learning rate searching space is $\{3e-4, 3e-5, 3e-6, 1e-5, 1e-6\}$.

where $B_{\partial G}^{(\tau)}, B_{\partial S}^{(\tau)} \subset R^3$ denote the border region of ground truth and segmentation surface at tolerance τ, which are defined as $B_{\partial G}^{(\tau)} = \{x \in R^3 \mid \exists \tilde{x} \in \partial G, \|x - \tilde{x}\| \leq \tau\}$ and $B_{\partial S}^{(\tau)} = \{x \in R^3 \mid \exists \tilde{x} \in \partial S, \|x - \tilde{x}\| \leq \tau\}$, respectively. In this paper, we use the default value $\tau = 1mm$ in the MICCAI Medical Segmentation Decathlon challenge evaluation code (http://medicaldecathlon.com/). The same setting is also used in a recent MICCAI endorsed COVID-19 infection segmentation challenge (https://covid-segmentation.grand-challenge.org/Evaluation/).

3.4. Ranking scheme

We use the similar ranking scheme in MICCAI 2018 Medical Segmentation Decathlon[8]. Specifically, a ranking score is assigned for each loss function based on the two metrics and the four segmentation tasks. Let l ($l \in \{1, \cdots, 20\}$) be the index of loss functions, t ($t \in \{1, \cdots, 4\}$) be the index of tasks, j ($j \in \{1, 2\}$) be the index of metrics, and k ($k \in \{1, \cdots, N_t\}$)[9] be the index of cases. All of the indices are independent to each other. The ranking scheme consists of the following two main steps.

Step 1. A significance score $score_{tj}(Loss_l)$ is separately determined for each loss function $Loss_l$, each task t and each metric $m_j \in \{DSC, NSD\}$, which is computed as follows:

- Performance assessment per case: determine performance[10] $m_j(Loss_l, case_{tk})$ of all loss functions $Loss_l$ for all test cases $case_{tk}$.
- Statistical tests: perform pairwise comparisons between all loss functions with the values $m_j(Loss_l, case_{tk}) - m_j(Loss_l', case_{tk})$ using Wilcoxon signed rank test[11].
- Significance scoring: $score_{tj}(Loss_l)$ equals the number of algorithms performing significantly worse than $Loss_l$ according to the statistical tests ($p = 0.05$).

Step 2. The final 'Rank Score' of each loss function is computed from the mean significance scores of all tasks and metrics, which is defined by

$$\text{Rank Score}(Loss_l) = \frac{1}{2 \times 4} \sum_{j=1}^{2} \sum_{t=1}^{4} score_{tj}(Loss_l).$$

4. Loss function comparison

In this section, we first present the segmentation results for each task in Figs. 2–8 with a violin plot that shows not only the summary statistics such as median and interquartile ranges, but also the entire distribution of the quantitative results. Then we show the ranking results of the 20 loss functions in Fig. 10. Table 2 summarizes the quantitative results of all 20 loss functions on the four segmentation tasks.

4.1. Single segmentation task

4.1.1. Mildly imbalanced segmentation

To investigate these loss functions on a *mildly imbalanced segmentation task*, we evaluate them on the liver segmentation dataset. As shown in Table 2, most of the loss functions (17/20) obtain highly accurate results with DSC above 0.90. DiceTopK loss achieves the best DSC and NSD, while pGDice loss obtains the lowest DSC, and TopK loss obtains the lowest performance in both DSC and NSD. Fig. 2 shows that the DSC values of TopK loss and pGDice

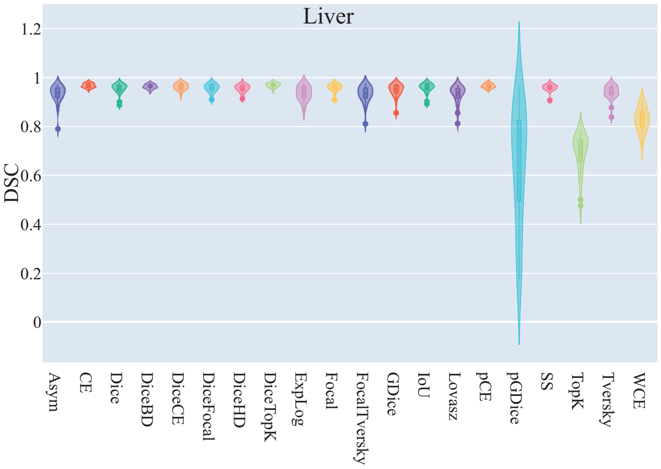
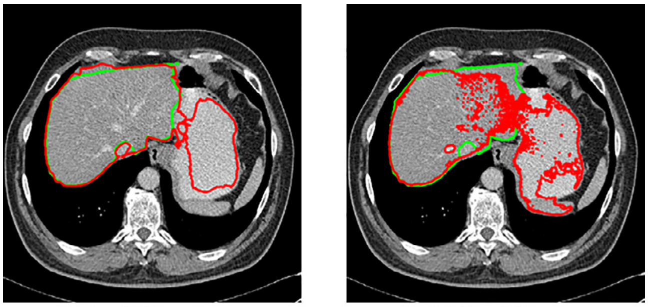


Fig. 2. Violin plot of the liver segmentation results with 20 loss functions.



(a) pGDice loss (b) TopK loss

Fig. 3. Failed segmentation cases of liver by pGDice loss and TopK loss.

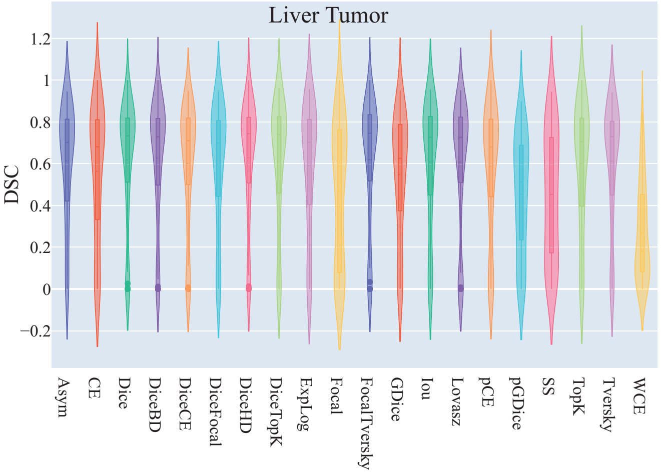
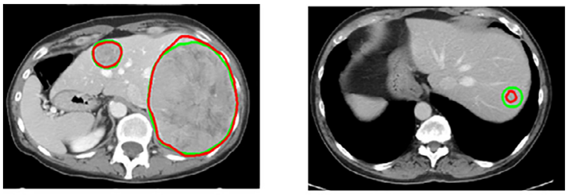


Fig. 4. Violin plot of the liver tumor segmentation results with 20 loss functions.



(a) Large tumor (b) Small tumor

Fig. 5. Segmentation results of large tumor and small tumor by Dice loss.

[8] http://medicaldecathlon.com/files/MSD-Ranking-scheme.pdf

[9] N_t is the number of cases in task t.

[10] In case of N/A value, set $m_j(Loss_l, case_{tk})$ to 0 for the cases with 0 predictions.

[11] scipy.stats.wilcoxon

Table 2
Average Dice similarity coefficient (DSC) and Normalized Surface Dice (NSD) of 20 loss functions with four datasets. Values in bold highlights the best results with a p-value less than 0.05, while underlined values point to the lowest results with a p-value less than 0.05.

Loss	Liver		Liver Tumor		Pancreas		Multi-organ	
	DSC	NSD	DSC	NSD	DSC	NSD	DSC	NSD
Asym	0.9315	0.6905	0.6134	0.4114	0.8234	0.6239	0.7526	0.6088
CE	0.9672	0.7962	0.5641	0.3715	0.8221	0.6321	0.8483	0.7200
Dice	0.9547	0.7598	0.6187	0.4291	0.8362	0.6688	0.8449	0.7136
DiceBD	0.9629	0.7702	0.6262	**0.4375**	0.8397	0.6713	0.8450	0.7105
DiceCE	0.9624	0.7743	0.6009	0.4114	0.8249	0.6298	0.8512	**0.7293**
DiceFocal	0.9566	0.7583	0.5951	0.4078	**0.8416**	**0.6721**	**0.8554**	**0.7339**
DiceHD	0.9556	0.7314	**0.6291**	**0.4390**	0.8408	0.6646	**0.8531**	0.7257
DiceTopK	**0.9690**	**0.8092**	0.6125	0.4208	0.8375	0.6598	**0.8512**	**0.7308**
ELL	0.9347	0.7254	0.5903	0.4047	0.8344	0.6508	0.8375	0.6689
Focal	0.9587	0.7528	0.4675	0.2781	0.8016	0.6034	0.8173	0.6642
FocalTversky	0.9320	0.6986	0.6193	0.4178	0.8229	0.6190	0.7497	0.6013
GDice	0.9474	0.7337	0.5486	0.3501	0.8285	0.6478	0.0132	0.0018
IoU	0.9568	0.7709	0.6079	0.4273	0.8353	0.6605	0.8439	0.7160
Lov*á*sz	0.9294	0.6639	0.6083	0.4205	0.8309	0.6521	0.6568	0.3845
pCE	0.9655	0.7852	0.5876	0.3829	0.8358	0.6580	0.8349	0.6967
pGDice	0.6449	0.3455	0.4526	0.2879	0.8156	0.6204	0.0595	0.0209
SS	0.9591	0.7571	0.4527	0.1941	0.7799	0.4781	0.7589	0.4958
TopK	0.6924	0.1073	0.5995	0.4051	0.8406	0.6709	0.8527	0.7323
Tversky	0.9390	0.6991	0.6120	0.4045	0.8260	0.6249	0.8371	0.6787
WCE	0.8284	0.2665	0.2697	0.0314	0.4744	0.0496	0.6904	0.2335

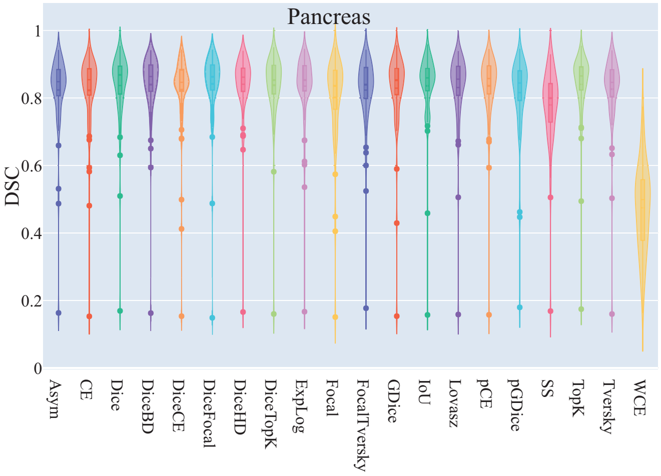


Fig. 6. Violin plot of the pancreas segmentation results with 20 loss functions.

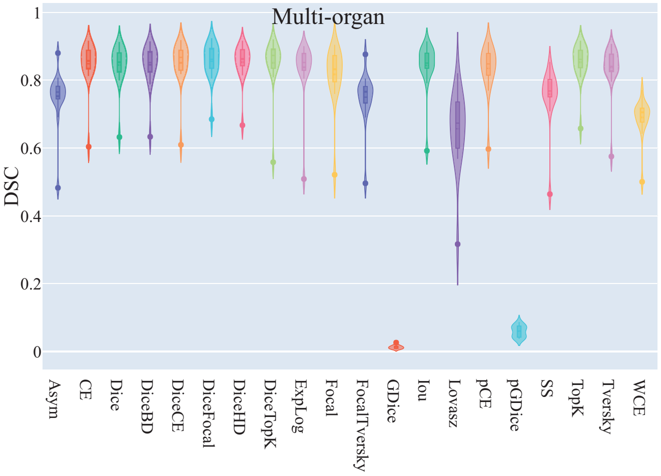
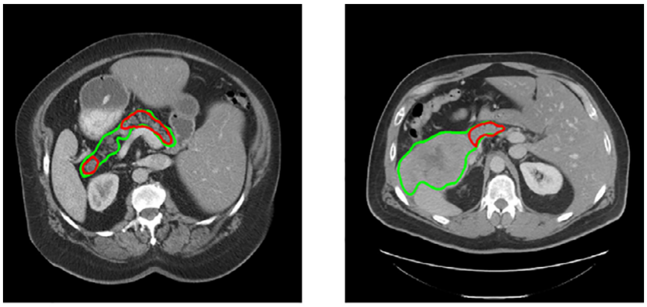


Fig. 8. Violin plot of the multi-organ segmentation results with 20 loss functions.



(a) Ambiguous boundary

(b) Large tumor

Fig. 7. Failed segmentation cases of pancreas by DiceFocal loss.

loss have significantly larger variations than the others. A further analysis of failed cases in Fig. 3 reveals that pGDice loss and TopK loss tend to generate over-segmented results. The possible reason may be that these two loss functions are designed for highly imbalanced segmentation tasks, which can lead to a bias on the foreground when they are applied to a more balanced task.

4.1.2. Highly imbalanced tumor segmentation

To investigate these loss functions on a *highly imbalanced tumor segmentation task*, we evaluate them on a large-scale liver tumor segmentation dataset that contains 400+ cases. As shown in Table 2, DiceHD loss achieves the best DSC, and DiceBD loss, and DiceHD loss achieve the best NSD, while weighted cross entropy obtains the worse DSC and NSD. It should be noted that some loss functions (e.g., Dice loss, DiceFocal loss, Tversky loss, and Asym loss) that are designed for highly imbalanced problems do not achieve the best performance, but they are comparable to the best-performing loss functions with a minor gap. However, some other loss functions (e.g., Focal loss, pGDice, SS loss, WCE) do not achieve satisfactory performance on this task.

Fig. 4 shows that the DSC values of the most loss functions have a bimodal distribution with modes near 0.8 and 0.1. In other words, the segmentation results are polarized, which is very different from the results in the liver segmentation task as shown in Fig. 2. Specifically, some tumor segmentation results are relatively good, while others are very poor. This is because the tumors have varying appearance, locations, shapes and sizes. As shown in

Fig. 5, the larger tumor with homogeneous appearance achieves a DSC with 0.85, while only a minor part of the small tumor is segmented.

4.1.3. Highly imbalanced organ segmentation

To investigate these loss functions in a *highly imbalanced organ segmentation task*, we evaluate the 20 loss functions on a large-scale pancreas dataset that contains 360+ cases. As shown in Table 2, DiceFocal loss and WCE achieve the best and worst performance, respectively. Similar to the results in liver tumor segmentation task, some loss functions (e.g., Dice BD loss, DiceTopK loss, ELL, pCE, and TopK) that are designed for highly imbalanced problems do not achieve the best performance, but they are comparable to the DiceFocal loss with a minor gap. However, some other loss functions (e.g., Focal loss, SS loss, and WCE) do not perform well on this task. Fig. 6 shows that the DSC values of results of all loss functions only have one assembling region near 0.8 rather than a polarized distribution. Only a few cases have very low DSC. Fig. 7 shows one case with an ambiguous boundary and another with a huge tumor in the pancreas. These challenging cases usually obtain lower performance.

4.1.4. Multi-class segmentation with both mildly and highly imbalanced labels

To investigate these loss functions on the multi-class segmentation task which contains *both mildly and highly imbalanced labels*, we evaluate the 20 loss functions on a multi-organ dataset that contains 8 labels. Some organs, such as the liver and the stomach, occupy a large volume, while some other organs, such as the gallbladder and esophagus, are much smaller. Table 2 shows the average DSC and NSD for each loss. DiceFocal loss, DiceHD loss, and DiceTopK loss achieve the best NSD, and DiceFocal loss, DiceCE loss, and DiceTopK loss achieve the best DSC. GDice loss obtains the lowest DSC and NSD and so does its variant pGDice loss.

Fig. 8 shows that the DSC distributions of most loss functions have one mode, while the medial DSC values of the loss functions have larger variances than the other tasks as shown in Figs. 2–6. In other words, some loss functions (e.g., Asym loss and FocalTversky loss) obtain comparable results to the top-performing loss function with a small gap in the binary segmentation tasks, but there exists a larger gap in the multi-organ segmentation where the labels are variously imbalanced. Fig. 9 presents visual examples of multi-organ segmentation results by DiceCE loss (DSC=0.91) and WCE (DSC=0.73), respectively. It can be found that DiceCE loss generates quite accurate results, while the results of WCE include some isolated outliers.

4.2. Different variants of TopK loss

There are two different variants of TopK Loss as introduced in Section 2.1.2. One variant keeps the $k\%$ percent worst pixels (termed as percentage variant) and the other one uses the pixels below a given threshold t (termed as threshold variant). We compare the two TopK loss variants with different percentages $k \in \{10\%, 30\%, 50\%, 70\%, 90\%\}$ and thresholds $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Table 3 presents the average DSC and NSD values on four segmentation tasks. Overall, the percentage variants of TopK loss can give reasonable results in all the experiments but the threshold variants could fail during the training process or generate bad segmentation results, indicating that the percentage variant is more robust than the threshold variant. The TopK-10% loss achieved the best results on liver tumor, pancreas, and multi-organ segmentation tasks but did not perform well on the liver segmentation task. However, the other TopK loss percentage variants (i.e., TopK-30%, TopK-50%, TopK-70%, and TopK-90%) achieved very high performance in terms of DSC on the liver segmentation task. The possible reason might be that only using the 10% pixels during training is not enough for the liver because liver is very large (the largest abdominal organ). Nevertheless, in this work, we mainly focus on the label-imbalanced tasks. Thus, we choose the TopK-10% loss as the default TopK loss because it achieved better performance on the challenging segmentation tasks.

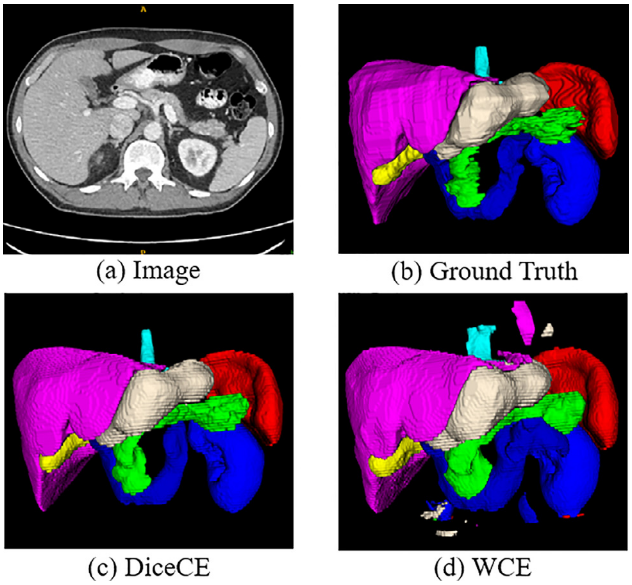


Fig. 9. Examples of multi-organ segmentation by DiceCE loss (c) and weighted cross entropy (WCE) (d), respectively.

4.3. Different variants of Dice loss

There are two different variants of Dice Loss (with and without squared terms in the denominator) as introduced in Section 2.2.2. Moreover, there are also two different implementations of Dice loss: sample Dice and batch Dice. Specifically, sample Dice computes the Dice loss for each sample in the minibatch independently and averages across the minibatch. Batch Dice treats the minibatch as a pseudo-volume and computes the Dice loss as if all voxels in the samples belonged to one training case (Isensee et al., 2021). Thus, there are in total four different Dice loss variants with the combination between w/wo squared terms and sample/batch Dice[12].

We compare the four Dice loss variants on four segmentation tasks where only plain Dice losses are used (without cross entropy). Table 4 presents the quantitative results. Overall the default setting achieved the best average DSC and NSD. BatchSquareDice loss outperformed the default Dice loss in liver segmentation tasks, but the improvements were marginal. SampleSquareDice achieved slightly better NSD and DSC than the default Dice loss in liver tumor and pancreas segmentation tasks, respectively, while the average DSC and NSD were inferior than Dice loss.

4.4. Rank Results

Table 2 summarizes the average DSC and NSD of 20 loss functions on four segmentation tasks. It can be found that none of

[12] The default Dice loss settings in nnU-Net (Isensee et al., 2019) are as follows: (1) the batch Dice is used for 3D full resolution (3d_fullres) model if 3D low resolution (3d_lowres) model exists; (2) If there is no 3d_lowres model, 3d_fullres model will use sample Dice; (3) 3d_lowres model always uses sample Dice. In our experiments, we always use 3d_fullres model and 3d_lowres model exists in all segmentation tasks. Thus, the batch Dice without square term is our default Dice loss.

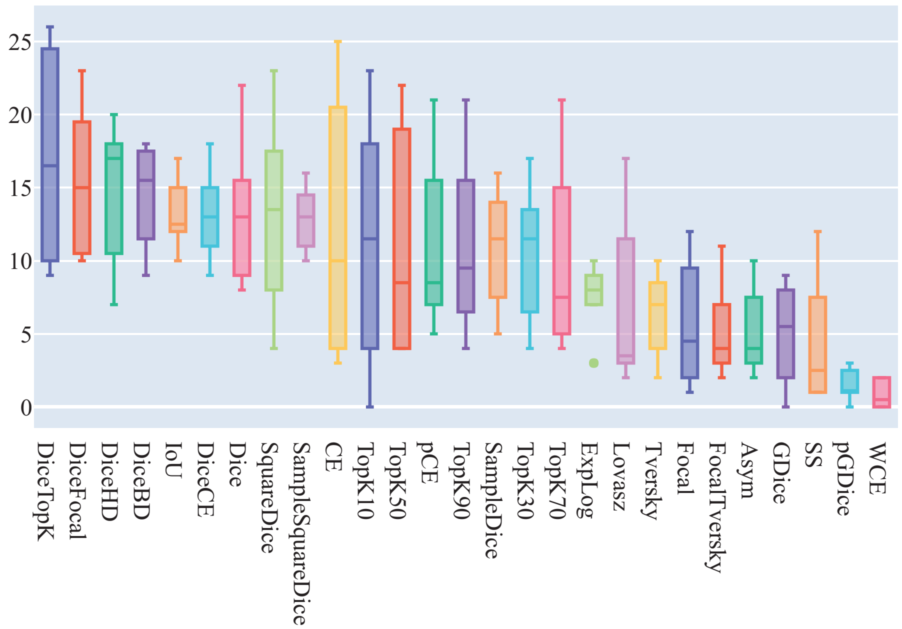


Fig. 10. Ranking results of 27 loss functions. From left to right, the loss functions are listed in descending order according to their mean ranking score.

Table 3
Average DSC and NSD of different TopK loss variants on four segmentation tasks. The “-” denotes that the results are not available, because the training process failed with these loss functions.

Loss	Liver		Liver Tumor		Pancreas		Multi-organ		Average	
	DSC	NSD	DSC	NSD	DSC	NSD	DSC	NSD	DSC	NSD
TopK-10%	0.6924	0.1073	**0.5995**	**0.4051**	**0.8406**	**0.6709**	**0.8527**	**0.7323**	0.7463	0.4789
TopK-Threshold (0.1)	0.0320	0.0303	-	-	0.0149	0.0082	-	-	0.0153	0.0096
TopK-30%	0.9621	0.7766	0.5212	0.3430	0.8350	0.6616	0.8420	0.7182	0.7901	0.6249
TopK-Threshold (0.3)	0.0348	0.0284	-	-	0.0026	0.0048	-	-	0.0094	0.0083
TopK-50%	0.9657	0.7881	0.5128	0.3237	0.8253	0.6508	0.8519	0.7241	0.7889	0.6217
TopK-Threshold (0.5)	0.0295	0.0272	-	-	0.0142	0.0082	0.7253	0.5530	0.1923	0.1471
TopK-70%	0.9648	0.7842	0.5644	0.3677	0.8301	0.6528	0.8422	0.7141	0.8004	0.6297
TopK-Threshold (0.7)	0.9090	0.6689	-	-	-	-	0.8432	0.7228	0.4381	0.3479
TopK-90%	**0.9649**	**0.7854**	0.5874	0.3967	0.8311	0.6475	0.8448	0.7119	**0.8071**	**0.6354**
TopK-Threshold (0.9)	0.9360	0.7129	0.4086	0.2335	0.7405	0.5057	0.8440	0.7232	0.7323	0.5438

Table 4
Average DSC and NSD of different Dice loss variants on four segmentation tasks.

Loss	Liver		Liver Tumor		Pancreas		Multi-organ		Average	
	DSC	NSD	DSC	NSD	DSC	NSD	DSC	NSD	DSC	NSD
Dice (Batch, no square)	0.9547	0.7598	**0.6187**	0.4291	0.8362	**0.6688**	**0.8449**	**0.7136**	**0.8136**	**0.6428**
BatchSquareDice	**0.9566**	**0.7633**	0.6101	0.4253	0.8427	0.6724	0.8418	0.7104	0.8128	**0.6428**
SampleDice (no square)	0.9345	0.7266	0.6033	0.4176	0.8392	0.6673	0.8390	0.7106	0.8040	0.6305
SampleSquareDice	0.9463	0.7418	0.6167	**0.4384**	**0.8388**	0.6674	0.8407	0.7095	0.8106	0.6393

the loss function can consistently achieve the best performance on all segmentation tasks, but we can identify the most robust loss functions based on the carefully designed ranking scheme. The results show that Dice-related compound loss functions perform quite robustly across different segmentation tasks. In addition, four groups out of the top-5 participants in recent MICCAI 2019 kidney and kidney tumor segmentation (KiTS) challenges employed compound loss functions (DiceCE or DiceTopK) (Heller et al., 2019), providing further evidence of the effectiveness of compound loss functions.

Some loss functions are not robust in different segmentation tasks. For example, GDice loss obtains promising results in binary segmentation tasks, but fails in multi-class segmentation. TopK loss achieves competitive results in liver tumor and pancreas segmentation compared with the top-performing loss funcitons, but it obtains the lowest score in liver segmentation. Weighted cross entropy often leads to a lower performance, which is consistent with the results in (Bertels et al., 2019; Eelbode et al., 2020).

Finally, we rank the 20 loss functions in Table 2 and the variants of Dice loss and TopK loss, in total 27 loss functions, based on the ranking scheme introduced in Section 3.4. As shown in Fig. 10, compound loss functions dominate the top places, such as DiceTopK, DiceFocal, DiceHD, and DiceBD. Excluding the compound loss functions, IoU loss, Dice loss, and cross entropy are the top-3 loss functions and this might be the reason for their popularity. It should be noted that the ranking results might be different if using different ranking schemes. In Section 5.1, we will discuss how the ranking changes when using different metrics.

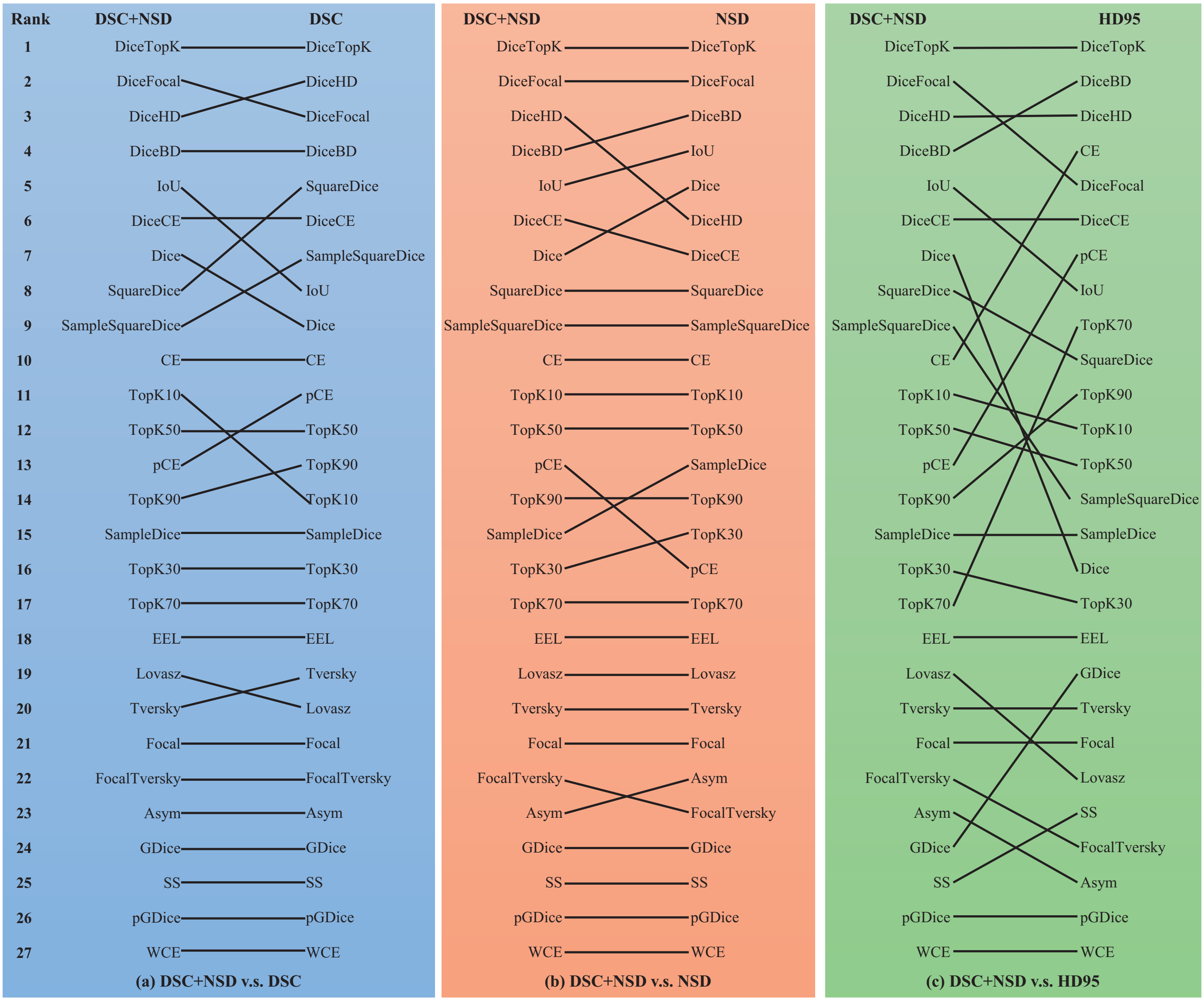


Fig. 11. Ranking stability analysis in terms of different metrics.

5. Discussion

5.1. Ranking stability

We analyze how the ranking changes when different metrics are used and the results are summarized in Fig. 11. We first compare the ranking results between the default DSC+NSD and only using DSC (Fig. 11 (a)) and only using NSD (Fig. 11 (b)), respectively. It can be found that the ranking results do not have too much fluctuations, and the DiceTopK loss ranks the 1st place among all three ranking schemes (DSC+NSD, DSC, NSD).

Moreover, we also include the popular 95% Hausdorff Distance (HD95)[13] in the ranking comparison (Fig. 11 (c)). When only using the HD95, the ranking has relatively large changes. We also found that boundary- and distance map-related losses can gain an advantage. In particular, the DiceBD loss ranks the second place and the pCE loss rises six places, which are their best ranking results for all metrics that we evaluated. In contrast, IoU loss and Dice loss drop three and nine places, respectively. In general, Dice-related compound losses are relatively robust, which have the best performance with all metrics in different ranking schemes.

There are two additional remarks for the cross entropy (CE) and the three loss functions with training tricks, including Lovász loss, BD loss, and HD loss. In particular, nnU-Net has a default setting of foreground regions oversampling[14] as mentioned in 3.2, which has an important impact on the performance of CE. We also evaluate the CE without using the oversampling on the four segmentation tasks, and the results show that the performance can degrade by 0.4%-6.1% in DSC and NSD. Specifically, without using the oversampling has more impact for highly label imbalanced task, for example, the DSC and NSD scores decrease by about 6% for the liver tumor segmentation but only drop by about 1% for the liver segmentation. Moreover, additional scheduling strategies are

[13] In the above Tables, we only present DSC and NSD scores and do not report HD95 scores because some cases will have “inf” HD95 when the segmentation results are empty. For example, in the liver tumor segmentation task, it is common that the tumors are completely missed by the model, especially for small tumors. In this situation, we can not compute the average HD95.

[14] 33% of the samples in a minibatch are guaranteed to contain at least one of the foreground classes.

applied to Lovász loss, BD loss, and HD loss during training, as introduced in 3.2, which can give these losses the best chance at succeeding. These training tricks are important to obtain better results (Ma et al., 2020). However, the other loss function do not require these training tricks and can be used in a plug-and-play way. Moreover, training with DiceBD loss or DiceHD loss is time-consuming, especially for the multi-class segmentation, because they need to compute distance transforms for each sample and each class during each iteration.

5.2. Loss function relationships

There are strong connections among existing loss functions as shown in Fig. 1. Most of the distribution-based and region-based loss functions are the variants of cross entropy and Dice loss, respectively. Boundary-based losses are motivated by minimizing the distance between two boundaries, but we show that they have similarities to Dice loss in formulation, as both of them are computed in a region-based way, and the key difference is the way the mismatched region is weighted. Moreover, Focal loss and TopK loss follow the same goal (focus on hard training examples) but they approach it quite differently. Different loss functions may respond differently to annotation errors. Focal loss and TopK for example will focus on mislabelled pixels, potentially learning useless information from mislabelled training cases. Dice loss weights false positives and false negatives equally, while asymmetric similarity loss and Tversky loss give different weights to the false positives and false negatives, which could achieve a better balance between precision and recall. Generalised Dice (GDice) loss extends the standard Dice loss by assigning different weights to different classes. The penalized GDice further extends GDice by assigning different weights to false positives and false negatives.

5.3. Limitations

There are some limitations in our work. First, we only use the suggested hyper-parameters in the original paper for each loss function, rather than applying extensive grid searching for all hyper-parameters. This is because that our main goal is to evaluate the plug-and-play loss functions and to recommend the most robust one to the community. Applying extensive hyper-parameters searching may improve the performance for some loss functions, however the best hyper-parameter for one task may not be the best for other tasks. Thus, the hyper-parameters obtained by searching cannot be a general recommendation.

Second, we have evaluated the 20 loss functions, and some of them achieved sate-of-the-art results, such as liver segmentation (Bilic et al., 2019) and multi-organ segmentation (Gibson et al., 2018). Nonetheless, we cannot claim that our implementations of loss functions are the exact same as the original proposed versions, because many of them are not open-source. Hence, we make all our code publicly available[15] for peer review. We have also tried our best to find the best learning rate for each loss function. In total, our experiments have cost more than six years GPU running time.

In the future, we will continue to add new general loss functions to our project and keep this work up to date (Shirokikh et al., 2020; Seo et al., 2021; Li et al., 2020). Moreover, exploring the loss landscape by visualization methods (Li et al., 2018a) may also help us to understand the geometric characteristics of the loss functions.

6. Conclusion

Recommendations for choosing loss functions: With 20 loss functions to choose from, it is important but hard to identify the one to try first when we deal with a new segmentation task. However, our results show that for the **QUESTION:** which loss function should we choose for medical image segmentation tasks? The **ANSWER** could be that, overall, using Dice-related compound loss functions is a better choice.

Summary: To the best of our knowledge, this work presents the first comprehensive review and empirical comparison of existing plug-and-play loss functions. Our loss function taxonomy provides an overview of the existing loss functions and presents a clear picture to assist in understanding the relationships among them. We also conduct a large set of experiments for 20 loss functions on four segmentation tasks with 6 public datasets from 10+ medical centers. Our code, dataset splits, and segmentation results are be publicly available and serve as a loss function benchmark, and we hope these results could greatly advance loss function development in the community.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jun Ma: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Jianan Chen:** Conceptualization, Methodology, Validation, Writing - original draft, Writing - review & editing. **Matthew Ng:** Conceptualization, Methodology. **Rui Huang:** Conceptualization, Methodology. **Yu Li:** Conceptualization, Methodology. **Chen Li:** Conceptualization, Methodology. **Xiaoping Yang:** Conceptualization, Methodology, Resources, Writing - original draft, Writing - review & editing, Funding acquisition. **Anne L. Martel:** Conceptualization, Methodology, Resources, Writing - original draft, Writing - review & editing, Funding acquisition.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 91630311, no. 11971229), and Nanjing University of Science and Technology PhD International Exchange Fellowship. We gratefully acknowledge Fabian Isensee for the public nnU-Net code and helpful discussions. We also thank Compute Canada (www.computecanada.ca) and Nanjing University High Performance Computing Center for the computational resource support.

References

Abraham, N., Khan, N. M., 2019. A novel focal tversky loss function with improved attention u-net for lesion segmentation. In: In 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE, pp. 683–687.

Bakas, S., Reyes, M., et Int, Menze, B., 2018. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. arXiv preprint arXiv:1811.02629.

Berman, M., Rannen Triki, A., Blaschko, M.B., 2018. 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, pp. 4413–4421.

Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.-A., Cetin, I., Lekadir, K., Camara, O., Ballester, M.A.G., et al., 2018. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved? IEEE Trans. Med. Imag. 37 (11), 2514–2525.

[15] https://github.com/JunMa11/SegLoss/tree/master/test

Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., Blaschko, M.B., 2019. Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 92–100.

Bilic, P., Christ, P.F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., Fu, C.-W., Han, X., Heng, P.-A., Hesser, J., et al., 2019. The liver tumor segmentation benchmark (lits). arXiv preprint arXiv:1901.04056.

Brosch, T., Yoo, Y., Tang, L.Y.W., Li, D.K.B., Traboulsee, A., Tam, R., 2015. Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention, pp. 3–11.

Brugnara, G., Isensee, F., Neuberger, U., Bonekamp, D., Petersen, J., Diem, R., Wildemann, B., Heiland, S., Wick, W., Bendszus, M., et al., 2020. Automated volumetric assessment with artificial neural networks might enable a more accurate assessment of disease burden in patients with multiple sclerosis. European Radiology 1–9.

Caliva, F., Iriondo, C., Martinez, A.M., Majumdar, S., Pedoia, V., 2019. Distance map loss penalty term for semantic segmentation. In: Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, pp. 1–5.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on Medical Image computing and computer-assisted intervention, pp. 424–432.

Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., Pal, C., 2016. The importance of skip connections in biomedical image segmentation. In: Deep Learning and Data Labeling for Medical Applications, pp. 179–187.

Eelbode, T., Bertels, J., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., Blaschko, M.B., 2020. Optimization for medical image segmentation: theory and practice when evaluating with dice score or jaccard index. IEEE Trans. Med. Imag. 39 (11), 3679–3690.

Fidon, L., Li, W., Garcia-Peraza-Herrera, L.C., Ekanayake, J., Kitchen, N., Ourselin, S., Vercauteren, T., 2017. Generalised wasserstein dice score for imbalanced multi–class segmentation using holistic convolutional networks. In: International MICCAI Brainlesion Workshop, pp. 64–76.

Ganaye, P.-A., Sdika, M., Triggs, B., Benoit-Cattin, H., 2019. Removing segmentation inconsistencies with semi-supervised non-adjacency constraint. Med. Image Analy. 58, 101551.

Gibson, E., Giganti, F., Hu, Y., Bonmati, E., Bandula, S., Gurusamy, K., Davidson, B., Pereira, S.P., Clarkson, M.J., Barratt, D.C., 2018. Automatic multi-organ segmentation on abdominal ct with dense v-networks. IEEE Trans. Med. Imag. 37 (8), 1822–1834.

Goyal, P., Kaiming, H., 2018. Focal loss for dense object detection. IEEE Trans. Pattern Analy. Mach. Intell. 39, 2999–3007.

Guan, S., Khan, A., Sikdar, S., Chitnis, P., 2019. Fully dense unet for 2d sparse photoacoustic tomography artifact removal. IEEE J. Biomed. Health Inform..

Hashemi, S.R., Mohseni Salehi, S.S., Erdogmus, D., Prabhu, S.P., Warfield, S.K., Gholipour, A., 2019. Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection. IEEE Access 7, 1721–1735.

Heller, N., Isensee, F., Maier-Hein, K.H., Hou, X., Xie, C., Li, F., Nan, Y., Mu, G., Lin, Z., Han, M., et al., 2019. The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge. Med. Image Analy. 67, 101821.

Hu, X., Li, F., Samaras, D., Chen, C., 2019. Topology-preserving deep image segmentation. In: Advances in Neural Information Processing Systems, pp. 5657–5668.

Ibtehaz, N., Rahman, M.S., 2020. Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. Neural Netw. 121, 74–87.

Isensee, F., Jäeger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H., 2021. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nat. Method. 18 (2), 203–211.

Isensee, F., Petersen, J., Kohl, S. A., Jäger, P. F., Maier-Hein, K. H., 2019. nnu-net: Breaking the spell on successful medical image segmentation. arXiv preprint arXiv:1904.08128.

Karimi, D., Salcudean, S.E., 2020. Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. IEEE Trans. Med. Imag. 39 (2), 499–513.

Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., Ayed, I.B., 2021. Boundary loss for highly unbalanced segmentation. Med. Image Analy. 67, 101851.

Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., Ben Ayed, I., 2019. Boundary loss for highly unbalanced segmentation. In: Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, 102, pp. 285–296.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR), pp. 1–15.

Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T., 2018. Visualizing the loss landscape of neural nets. In: Advances in Neural Information Processing Systems, pp. 6389–6399.

Landman, B., Xu, Z., Igelsias, J., Styner, M., Langerak, T., Klein, A. https://www.synapse.org/#!Synapse:syn3193805/wiki/217789.

Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.-W., Heng, P.-A., 2018. H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. IEEE Trans. Med. Imag. 37 (12), 2663–2674.

Li, Z., Kamnitsas, K., Glocker, B., 2020. Analyzing overfitting under class imbalance in neural networks for image segmentation. IEEE Trans. Med. Imag..

Ma, J., Wei, Z., Zhang, Y., Wang, Y., Lv, R., Zhu, C., Chen, G., Liu, J., Peng, C., Wang, L., Wang, Y., Chen, J., 2020. How distance transform maps boost segmentation cnns: an empirical study. In: Medical Imaging with Deep Learning. In: Proceedings of Machine Learning Research, 121, pp. 479–492.

Menze, B.H., et Int, Leemput, K.V., 2015. The multimodal brain tumor image segmentation benchmark (brats). IEEE Trans. Med. Imag. 34 (10), 1993–2024.

Milletari, F., Navab, N., Ahmadi, S.-A., 2016. V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571.

Nikolov, S., Blackwell, S., Mendes, R., De Fauw, J., Meyer, C., Hughes, C., Askham, H., Romera-Paredes, B., Karthikesalingam, A., Chu, C., et al., 2018. Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy. arXiv preprint arXiv:1809.04430.

Rahman, M.A., Wang, Y., 2016. Optimizing intersection-over-union in deep neural networks for image segmentation. In: International symposium on visual computing, pp. 234–244.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention, pp. 234–241.

Salehi, S.S.M., Erdogmus, D., Gholipour, A., 2019. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In: International Workshop on Machine Learning in Medical Imaging, pp. 379–387.

Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., Rueckert, D., 2019. Attention gated networks: Learning to leverage salient regions in medical images. Med. Image Analy. 53, 197–207.

Seo, H., Bassenne, M., Xing, L., 2021. Closing the gap between deep neural network modeling and biomedical decision-making metrics in segmentation via adaptive loss functions. IEEE Trans. Med. Imag. 40 (2), 585–593.

Shirokikh, B., Shevtsov, A., Kurmukov, A., Dalechina, A., Krivov, E., Kostjuchenko, V., Golanov, A., Belyaev, M., 2020. Universal loss reweighting to balance lesion size inequality in 3d medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 523–532.

Simpson, A. L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., van Ginneken, B., Kopp-Schneider, A., Landman, B. A., Litjens, G., Menze, B., et al., 2019. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063.

Su, Y., Jihoon, K., Young-Hak, K., 2019. Major vessel segmentation on x-ray coronary angiography using deep networks with a novel penalty loss function. In: Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, pp. 1–5.

Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Cardoso, M.J., 2017. 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, pp. 240–248.

Taghanaki, S.A., Zheng, Y., Zhou, S.K., Georgescu, B., Sharma, P., Xu, D., Comaniciu, D., Hamarneh, G., 2019. Combo loss: Handling input and output imbalance in multi-organ segmentation. Comput. Med. Imag. Graph. 75, 24–33.

Wong, K.C., Moradi, M., Tang, H., Syeda-Mahmood, T., 2018. 3d segmentation with exponential logarithmic loss for highly unbalanced object sizes. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 612–619.

Wu, Z., Shen, C., Hengel, A. v. d., 2016. Bridging category-level and instance-level semantic image segmentation. arXiv preprint arXiv:1605.06885.

Xiao, X., Lian, S., Luo, Z., Li, S., 2018. Weighted res-unet for high-quality retina vessel segmentation. In: 2018 9th International Conference on Information Technology in Medicine and Education (ITME), pp. 327–331.

Zhu, W., Huang, Y., Zeng, L., Chen, X., Liu, Y., Qian, Z., Du, N., Fan, W., Xie, X., 2019. Anatomynet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. Med. Phys. 46 (2), 576–589.

Zhuang, X., Li, L., Payer, C., Štern, D., Urschler, M., Heinrich, M.P., Oster, J., Wang, C., Smedby, Ö., Bian, C., et al., 2019. Evaluation of algorithms for multi-modality whole heart segmentation: An open-access grand challenge. Med. Image Analy. 58, 101537.