# TriMix: A General Framework for Medical Image Segmentation from Limited Supervision

Zhou Zheng[1,*] Yuichiro Hayashi[1] Masahiro Oda[1]
Takayuki Kitasaka[2] Kensaku Mori[1,3,*]

[1]Nagoya University [2]Aichi Institute of Technology
[3]National Institute of Informatics
[*]zzheng@mori.m.is.nagoya-u.ac.jp, kensaku@is.nagoya-u.ac.jp

**Abstract.** We present a general framework for medical image segmentation from limited supervision, reducing the reliance on fully and densely labeled data. Our method is simple, jointly trains *tri*ple diverse models, and adopts a *mix* augmentation scheme, and thus is called *TriMix*. TriMix imposes consistency under a more challenging perturbation, *i.e.*, combining data augmentation and model diversity on the tri-training framework. This straightforward strategy enables TriMix to serve as a strong and general learner learning from limited supervision using different kinds of imperfect labels. We conduct extensive experiments to show TriMix's generic purpose for semi- and weakly-supervised segmentation tasks. Compared to task-specific state-of-the-arts, TriMix achieves competitive performance and sometimes surpasses them by a large margin. The code is available at https://github.com/MoriLabNU/TriMix.

## 1 Introduction

Segmentation is fundamental in medical image analysis, recognizing anatomical structures. Supervised learning has led to a series of advancements in medical image segmentation [1]. However, the availability of fully and densely labeled data is a common bottleneck in supervised learning, especially in medical image segmentation, since annotating pixel-wise labels is usually tedious and time-consuming and requires expert knowledge. Thus, training a model with limited supervision using datasets with imperfect labels is essential.

Existing works have made efforts to take advantage of unlabeled data and weakly labeled data to train segmentation models [2] with semi-supervised learning (SSL) [3–5] and weakly-supervised learning [6–8]. Semi-supervised segmentation [9–11] is an effective paradigm for learning a model from scarce annotations, exploiting labeled and unlabeled data. Weakly-supervised segmentation aims to alleviate the longing for densely labeled data, utilizing sparse annotations, *e.g.*, points and scribbles, as supervision signals [2]. In this study, in addition to semi-supervised segmentation, we focus on *scribble-supervised* segmentation, one of the hottest topics in the family of weakly-supervised learning. A conceptual comparison of fully-supervised, semi-supervised, and scribble-supervised segmentation is shown in Fig. 1.

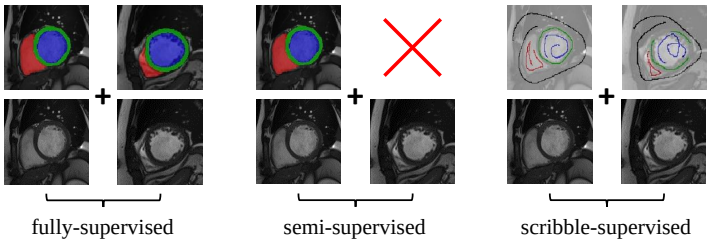


**Fig. 1. Conceptual comparison** of fully-supervised (using fully and densely labeled data), semi-supervised (using a part of densely labeled data and unlabeled data), and scribble-supervised (using data with scribble annotations) segmentation.

Consistency regularization aims to enforce the prediction agreement under different kinds of perturbations, *e.g.*, input augmentation [3, 9], network diversity [11, 12], and feature perturbation [13]. Recent works [7, 8, 14–20] involving consistency regularization shows advanced performance tackling limited supervision. Despite their success in learning from non-fullness supervision, an impediment is that existing studies are *task-specific* for semi- and scribble-supervised segmentation. Driven by this limitation, a question to ask is: *Does a framework generic to semi- and scribble-supervised segmentation exist*? Although the two tasks leverage different kinds of imperfect labels, indeed, they have the same intrinsic goal: mining the informative information as much as possible from pixels with no ground truth. Thus, such a framework should *exist* once it can excellently learn representations from the unlabeled pixels.

Consistency regularization under a more rigorous perturbation empirically leads to an improved generalization [11]. However, lacking sufficient supervision, models may output inaccurate predictions and then learn from these under consistency enforcement. This vicious cycle would accumulate prediction mistakes and finally lead to degraded performance. Thus, the key to turning the vicious cycle into a virtuous circle is increasing the quality of model outputs when adopting a more challenging consistency regularization. From these perspectives, we hypothesize that an eligible framework should be endowed with these characteristics: **(i)** it should output *more accurate predictions*, and **(ii)** it should be trained with consistency regularization under a *more challenging perturbation*.

Based on the above hypothesis, we find a solution: we present a general and effective framework that, for the first time, shows its dual purpose for both semi- and scribble-supervised segmentation tasks. The method is simple, jointly trains *tri*ple models, and adopts a *mix* augmentation scheme, and thus is called *TriMix*. To meet the requirement of **(i)**, TriMix maintains triple networks, which have identical structures but different initialization to introduce *model perturbation* and imposes consistency to minimize disagreement among models, inspired by the original tri-training strategy [21]. Intuitively, more diverse models can extract more informative information from the dataset. Each model receives valu-

able information from the other two through intra-model communication and then generates more accurate predictions. To meet the requirement of **(ii)**, the model diversity is further blended with *data perturbation*, which accompanies the mix augmentation scheme, to form a more challenging perturbation. We hypothesize that the tri-training scheme within TriMix well complements consistency regularization under the hybrid perturbation. This *self-complementary* manner enables TriMix to serve as a general learner learning from limited supervision using different kinds of imperfect labels. Our contributions are:

- We propose a simple and effective method called TriMix and show its generic solution for semi- and scribble-supervised segmentation for the first time.
- We show that purely imposing consistency under a more challenging perturbation, *i.e.*, combining data augmentation and model diversity, on the tri-training framework can be a general mechanism for limited supervision.
- We first validate TriMix on the semi-supervised task. TriMix presents competitive performance against state-of-the-art (SOTA) methods and surprisingly strong potential under the one-shot setting[1], which is rarely challenged by existing semi-supervised segmentation methods.
- We then evaluate TriMix on the scribble-supervised task. TriMix surpasses the mainstream methods by a large margin and realizes new SOTA performance on the public benchmarks.

## 2 Related Work

**Semi-supervised learning (SSL)** trains a model utilizing both labeled and unlabeled data. Existing SSL methods are generally based on pseudo-labeling (also called self-training) [5, 25–27] and consistency regularization [3, 4, 28, 29]. Pseudo-labeling takes the model's class prediction as a label to train against, but the label quality heavily influences the performance. **Consistency regularization** assumes predictions should be invariant under perturbations, such as input augmentation [3, 9], network diversity [11, 12], and feature perturbation [13]. Consistency regularization usually performs better than self-training and has been widely involved in the task of **semi-supervised segmentation** [14–18, 30, 31]. A more challenging perturbation empirically profits model generalization if the model could sustainably generate accurate predictions [11]. In this work, we introduce a hybrid perturbation harsher than its elements, *i.e.*, data augmentation, and model diversity.

**Weakly-supervised segmentation** learns a model using the dataset with weak annotations, *e.g.*, bounding boxes, scribbles, sparse dots, and polygons [2]. In this work, we utilize scribbles as weak annotations, which are mostly used in computer vision community, from classical methods [32, 33] to current **scribble-supervised** methods [6–8, 19, 34–37], due to the convenient format. To learn

[1] Note that the concepts of one-shot learning [22–24] and semi-supervised learning should be *different*. We *borrow* the phrase "one-shot" to define a more challenging semi-supervised setting where only one labeled sample is available during training.

from scribble supervision, some methods [34–36] make efforts to construct complete labels based on the scribble for training. Other works like [37, 38] explore possible losses to regularize the training from scribble annotations, and the scheme of [6] adds additional modules to improve the segmentation accuracy. Recently, consistency regularization is explored in several works [7, 8, 20, 39].

**Data augmentation** generates virtual training examples to improve model generalization. Auto-augmentation methods [40–43] automatically search for optimal data augmentation policies and show higher accuracy than handmade schemes but with relatively higher search costs. In our study, we focus on the **mix augmentation** [44–50], which is one type of strong data augmentation and is more efficient than auto-augmentation methods. Mix augmentation mixes two inputs and the corresponding labels up in some way to create virtual samples for training. It has been widely applied in semi-supervised segmentation [9–11] as an effective way to import data perturbation and synthesize new samples during training. In [7], mix segmentation is firstly introduced to increment supervision for scribble-supervised segmentation.

**Co-training and tri-training** are two SSL approaches in a similar flavor, which maintain multiple models and regularize the disagreement among the outputs of models. Co-training framework [51, 52] assumes there are sufficient and different views of the training data, each of which can independently train a model. Maintaining view diversity, in some sense, is similar to the data perturbation in SSL. Co-training has been extended to semi-supervised segmentation [18, 53]. Unlike co-training, tri-training [21] does not require view difference. Instead, it introduces model diversity and minimizes the disagreement among various outputs. This strategy is similar to imposing consistency under the model perturbation in SSL. There are several variants of tri-training [54–57], but *none* are for semi- or scribble-supervised segmentation. In this work, we revisit tri-training and explore its potential and general solution for handling limited supervision when it meets mix augmentation.

## 3 Method

### 3.1 Overview

This paper proposes a simple and general framework, TriMix, to tackle semi- and scribble-supervised segmentation. The plain architecture of TriMix is illustrated in Fig. 2. TriMix adheres to the spirit of tri-training, simultaneously learning triple networks $f_1$, $f_2$, and $f_3$, which have identical structures but different weights $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$, to import network inconsistency. In addition, mix augmentation is adopted to introduce input data perturbation. Generally, assume a mini-batch $\mathbf{b} = \{\mathbf{x}, \mathbf{y}\}$ is fetched at each training iteration, where $\mathbf{x}$ and $\mathbf{y}$ are images and the corresponding ground truth. TriMix involves three steps to process a batch flow at each training iteration.

**Step 1: first forward pass**. For $i \in \{1, 2, 3\}$, each network $f_i$ is fed with images $\mathbf{x}$ and outputs the prediction $\mathbf{p}_i$. A supervised loss $L_{sup}(\mathbf{p}_i, \mathbf{y})$ is then imposed between $\mathbf{p}_i$ and the ground truth $\mathbf{y}$.

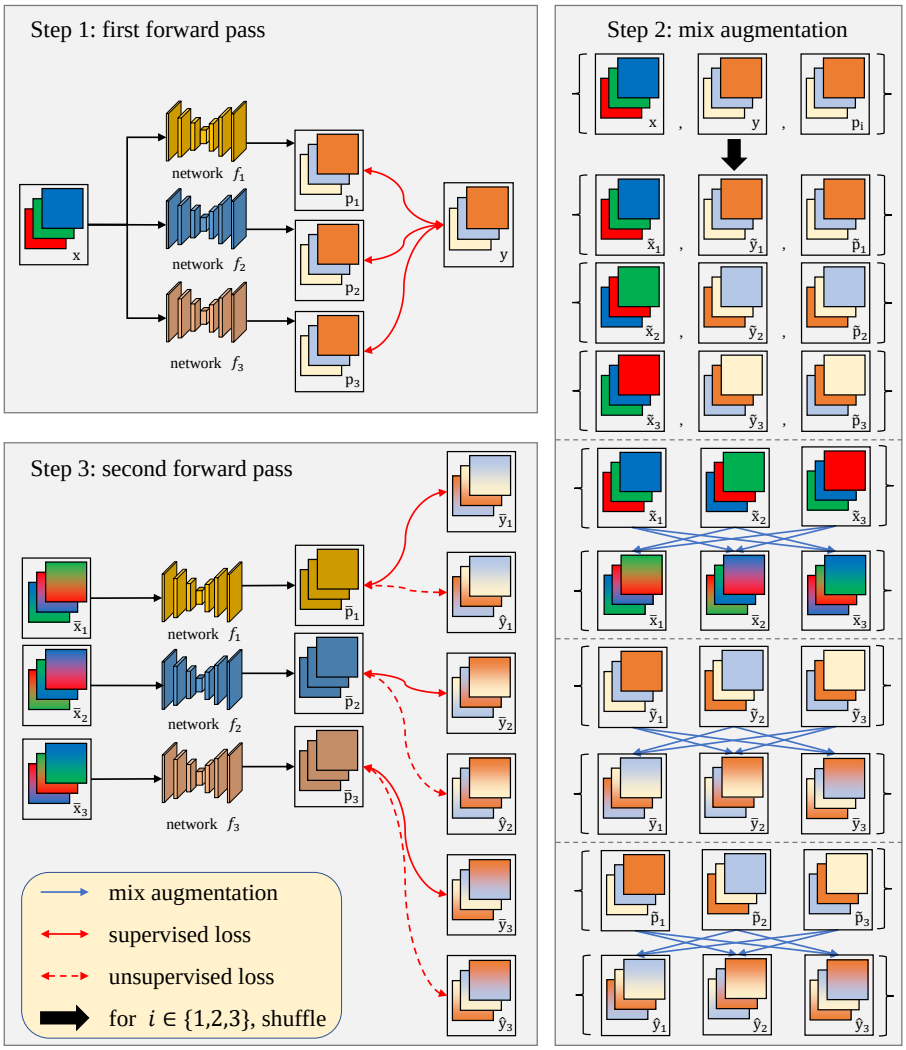


**Fig. 2. Overview of TriMix**. TriMix maintains triple networks $f_1$, $f_2$, and $f_3$, which have same architectures but different weights. Three steps are taken when given a mini-batch containing images $\mathbf{x}$ and ground truth $\mathbf{y}$ at each training iteration. **Step 1: first forward pass.** For $i \in \{1, 2, 3\}$, each network $f_i$ outputs $\mathbf{p}_i$ for $\mathbf{x}$, with the supervision of $\mathbf{y}$. **Step 2: mix augmentation.** Three batches $\{\mathbf{x}, \mathbf{y}, \mathbf{p}_1\}$, $\{\mathbf{x}, \mathbf{y}, \mathbf{p}_2\}$, and $\{\mathbf{x}, \mathbf{y}, \mathbf{p}_3\}$ are randomly shuffled to obtain new batches $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{y}}_1, \tilde{\mathbf{p}}_1\}$, $\{\tilde{\mathbf{x}}_2, \tilde{\mathbf{y}}_2, \tilde{\mathbf{p}}_2\}$, and $\{\tilde{\mathbf{x}}_3, \tilde{\mathbf{y}}_3, \tilde{\mathbf{p}}_3\}$. Then each pair of these new batches are mixed up to form batches $\{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3\}$, $\{\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2, \bar{\mathbf{y}}_3\}$, $\{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \hat{\mathbf{y}}_3\}$. Squares with mixed colors indicate mixed samples. **Step 3: second forward pass.** For $i \in \{1, 2, 3\}$, each network $f_i$ outputs $\bar{\mathbf{p}}_i$ for $\bar{\mathbf{x}}_i$, with the supervision of $\bar{\mathbf{y}}_i$. An unsupervised loss is calculated between $\bar{\mathbf{p}}_i$ and $\hat{\mathbf{y}}_i$. Note that $\hat{\mathbf{y}}_i$ can be soft (probability maps) or hard pseudo-labels (one-hot maps).

**Step 2: mix augmentation.** With Step 1, we obtain three batches $\mathbf{b}_1 = \{\mathbf{x}, \mathbf{y}, \mathbf{p}_1\}$, $\mathbf{b}_2 = \{\mathbf{x}, \mathbf{y}, \mathbf{p}_2\}$, and $\mathbf{b}_3 = \{\mathbf{x}, \mathbf{y}, \mathbf{p}_3\}$. The goal is to mix up the pair of $(\mathbf{b}_2, \mathbf{b}_3)$, the pair of $(\mathbf{b}_1, \mathbf{b}_3)$, and the pair of $(\mathbf{b}_1, \mathbf{b}_2)$ to generate new batches. Similar to the mixing operation described in original papers [44, 46], we first randomly shuffle $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$ to generate three new batches of $\tilde{\mathbf{b}}_1 = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{y}}_1, \tilde{\mathbf{p}}_1\}$, $\tilde{\mathbf{b}}_2 = \{\tilde{\mathbf{x}}_2, \tilde{\mathbf{y}}_2, \tilde{\mathbf{p}}_2\}$, and $\tilde{\mathbf{b}}_3 = \{\tilde{\mathbf{x}}_3, \tilde{\mathbf{y}}_3, \tilde{\mathbf{p}}_3\}$, in which $\tilde{\mathbf{x}}_1$, $\tilde{\mathbf{x}}_2$, and $\tilde{\mathbf{x}}_3$ have different image order, and each $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{p}}_i$ correspond to $\tilde{\mathbf{x}}_i$ for $i \in \{1, 2, 3\}$. Afterward, we apply the mix augmentation to the pair of $\left(\tilde{\mathbf{b}}_2, \tilde{\mathbf{b}}_3\right)$, the pair of $\left(\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_3\right)$, and the pair of $\left(\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2\right)$ to generate new batches of $\bar{\mathbf{b}}_1 = \{\bar{\mathbf{x}}_1, \bar{\mathbf{y}}_1, \hat{\mathbf{y}}_1\}$, $\bar{\mathbf{b}}_2 = \{\bar{\mathbf{x}}_2, \bar{\mathbf{y}}_2, \hat{\mathbf{y}}_2\}$, and $\bar{\mathbf{b}}_3 = \{\bar{\mathbf{x}}_3, \bar{\mathbf{y}}_3, \hat{\mathbf{y}}_3\}$ with mixed samples. Take the pair of $\left(\tilde{\mathbf{b}}_2, \tilde{\mathbf{b}}_3\right)$, for example. Each image of $\tilde{\mathbf{x}}_2$ is mixed with the image indexed in the same order in $\tilde{\mathbf{x}}_3$ to yield $\bar{\mathbf{x}}_1$, then $\tilde{\mathbf{y}}_2$ and $\tilde{\mathbf{y}}_3$, $\tilde{\mathbf{p}}_2$ and $\tilde{\mathbf{p}}_3$ are proportionally mixed to get $\bar{\mathbf{y}}_1$ and $\hat{\mathbf{y}}_1$. Squares with mixed colors in Fig. 2 indicate mixed samples.

**Step 3: second forward pass.** For $i \in \{1, 2, 3\}$, we feed each network $f_i$ with mixed images $\bar{\mathbf{x}}_i$ to get the individual prediction $\bar{\mathbf{p}}_i$. Each $\bar{\mathbf{p}}_i$ is optimized to be close to the mixed ground truth $\bar{\mathbf{y}}_i$ with a supervised loss $L_{sup}\left(\bar{\mathbf{p}}_i, \bar{\mathbf{y}}_i\right)$. Besides, consistency is enforced between $\bar{\mathbf{p}}_i$ and the mixed pseudo-labels $\hat{\mathbf{y}}_i$, with an unsupervised loss $L_{unsup}\left(\bar{\mathbf{p}}_i, \hat{\mathbf{y}}_i\right)$. Note that $\hat{\mathbf{y}}_i$ could be soft (probability maps) or hard pseudo-labels (one-hot maps). A typical choice selected by most methods [4, 14, 17] is a soft pseudo-label, and an unsupervised loss $L^p_{unsup}$ compares the *probability consistency* by the mean square error (MSE) equation. By contrast, several works, *e.g.*, [8, 10] utilize a hard pseudo-label, where an unsupervised loss $L^s_{unsup}$ calculates the *pseudo supervision consistency*.

To conclude, the total optimization objective of each network is

$$L_i = L_{sup}\left(\mathbf{p}_i, \mathbf{y}\right) + \lambda_1 L_{sup}(\bar{\mathbf{p}}_i, \bar{\mathbf{y}}_i) + \lambda_2 L_{unsup}\left(\bar{\mathbf{p}}_i, \hat{\mathbf{y}}_i\right), \tag{1}$$

where $i \in \{1, 2, 3\}$ is the index pointing out items corresponding to network $f_i$, and $\lambda_1$ and $\lambda_2$ are hyperparameters to balance each term.

**Default settings.** In this study, we adopt *pseudo supervision consistency*. We will show that TriMix potentially achieves better accuracy integrated with pseudo supervision consistency than probability consistency in Section 4.4. Besides, we utilize *CutMix* [46] as the mix strategy, similar to [9–11], but note that other kinds of mix augmentations should also fit our framework.

**Inference process.** Triple networks with different weights are in TriMix. For a test sample, each network individually outputs a prediction. We will report the *average* result of them and report their *ensemble* result obtained by soft voting.

The below two sections will show how TriMix can be applied to semi- and scribble-supervised tasks, following the standard process from Step 1 to Step 3.

### 3.2 TriMix in Semi-Supervised Segmentation

Semi-supervised segmentation aims to learn a model by exploiting two given datasets: labeled dataset $\mathbf{D}_l = \{\mathbf{X}_l, \mathbf{Y}_l\}$, and unlabeled dataset $\mathbf{D}_u = \{\mathbf{X}_u\}$, where $\mathbf{X}$ and $\mathbf{Y}$ are images and the corresponding ground truth.

Assume a mini-batch of labeled data $\mathbf{b}_l = \{\mathbf{x}_l, \mathbf{y}_l\} \in \mathbf{D}_l$ and a mini-batch of unlabeled data $\mathbf{b}_u = \{\mathbf{x}_u\} \in \mathbf{D}_u$ are sampled at each training iteration. We illustrate the training detail of $\mathbf{b}_l$ and $\mathbf{b}_u$ in the following.

First, the mini-batch $\mathbf{b}_l$ contains the images and the corresponding ground truth, and TriMix can be optimized with $\mathbf{b}_l$ obeying the standard process as illustrated in Fig. 2. However, existing SSL methods, *e.g.*, [10, 11] *rarely* introduce perturbations to the labeled data, even though it is beneficial for performance. Following previous methods, we optimize TriMix *only with Step 1* and eliminate the processes of Step 2 and Step 3 when using $\mathbf{b}_l$. Thus, for $i \in \{1, 2, 3\}$, assume each network $f_i$ outputs perdition $\mathbf{p}_{l_i}$ for images $\mathbf{x}_l$, then only a supervised loss $L_{sup}\left(\mathbf{p}_{l_i}, \mathbf{y}_l\right)$ is calculated between $\mathbf{p}_{l_i}$ and the ground truth $\mathbf{y}_l$.

Second, the mini-batch $\mathbf{b}_u$ contains images $\mathbf{x}_u$ but no related labels. TriMix can still be optimized with $\mathbf{b}_u$ following the standard process as illustrated in Fig. 2 but *without supervised terms*. Specifically, for $i \in \{1, 2, 3\}$, each network $f_i$ outputs individual prediction $\mathbf{p}_{u_i}$ for $\mathbf{x}_u$ with the first forward pass at Step 1. There is no supervised term at Step 1 for each $\mathbf{p}_{u_i}$, due to the lack of ground truth. At Step 2, three batches $\mathbf{b}_{u_1} = \left\{\mathbf{x}_u, \mathbf{p}_{u_1}\right\}$, $\mathbf{b}_{u_2} = \left\{\mathbf{x}_u, \mathbf{p}_{u_2}\right\}$, and $\mathbf{b}_{u_3} = \left\{\mathbf{x}_u, \mathbf{p}_{u_3}\right\}$, which contain no ground truth, can be mixed up to generate augmented batches $\bar{\mathbf{b}}_{u_1} = \{\bar{\mathbf{x}}_{u_1}, \hat{\mathbf{y}}_{u_1}\}$,$\bar{\mathbf{b}}_{u_2} = \{\bar{\mathbf{x}}_{u_2}, \hat{\mathbf{y}}_{u_2}\}$, and $\bar{\mathbf{b}}_{u_3} = \{\bar{\mathbf{x}}_{u_3}, \hat{\mathbf{y}}_{u_3}\}$, that have no mixed ground truth. At Step 3, each network $f_i$ fed with mixed images $\bar{\mathbf{x}}_{u_i}$ is expected to output a similar prediction $\bar{\mathbf{p}}_{u_i}$ compared to $\hat{\mathbf{y}}_{u_i}$, with an unsupervised loss $L_{unsup}\left(\bar{\mathbf{p}}_{u_i}, \hat{\mathbf{y}}_{u_i}\right)$.

To conclude, the total training objective of each network in this task is

$$L_i = L_{sup}\left(\mathbf{p}_{l_i}, \mathbf{y}_l\right) + \lambda L_{unsup}(\bar{\mathbf{p}}_{u_i}, \hat{\mathbf{y}}_{u_i}), \tag{2}$$

where items with $i \in \{1, 2, 3\}$ correspond to network $f_i$, and $\lambda$ is a trade-off hyperparameter. Moreover, we use the *dice* loss [58] $L_{dice}$ as both the supervised and unsupervised losses. Thus, Eq. (2) is re-written as

$$L_i = \underbrace{L_{dice}\left(\mathbf{p}_{l_i}, \mathbf{y}_l\right)}_{\text{sup}} + \underbrace{\lambda L_{dice}\left(\bar{\mathbf{p}}_{u_i}, \hat{\mathbf{y}}_{u_i}\right)}_{\text{unsup}}. \tag{3}$$

### 3.3 TriMix in Scribble-Supervised Segmentation

Scribble-supervised segmentation trains a model from a given dataset $\mathbf{D}_s = \{\mathbf{X}_s, \mathbf{Y}_s\}$, where $\mathbf{X}_s$ and $\mathbf{Y}_s$ are images and the related scribble annotations.

Let $\mathbf{b}_s = \{\mathbf{x}_s, \mathbf{y}_s\} \in \mathbf{D}_s$ indicate a mini-batch fetched at every training iteration. Since $\mathbf{b}_s$ contains images and the corresponding ground truth in scribbles, we follow the standard process illustrated in Fig. 2 to train TriMix with $\mathbf{b}_s$. Let us say, for $i \in \{1, 2, 3\}$, each network $f_i$ outputs its prediction $\mathbf{p}_{s_i}$ for $\mathbf{x}_s$ at Step 1, and we obtain mixed batches of $\bar{\mathbf{b}}_{s_1} = \{\bar{\mathbf{x}}_{s_1}, \bar{\mathbf{y}}_{s_1}, \hat{\mathbf{y}}_{s_1}\}$, $\bar{\mathbf{b}}_{s_2} = \{\bar{\mathbf{x}}_{s_2}, \bar{\mathbf{y}}_{s_2}, \hat{\mathbf{y}}_{s_2}\}$, and $\bar{\mathbf{b}}_{s_3} = \{\bar{\mathbf{x}}_{s_3}, \bar{\mathbf{y}}_{s_3}, \hat{\mathbf{y}}_{s_3}\}$ at Step 2. Then identical to Eq. (1), the training objective of each network $f_i$ in scribble-supervised segmentation is

$$L_i = L_{sup}\left(\mathbf{p}_{s_i}, \mathbf{y}_s\right) + \lambda_1 L_{sup}\left(\bar{\mathbf{p}}_{s_i}, \bar{\mathbf{y}}_{s_i}\right) + \lambda_2 L_{unsup}\left(\bar{\mathbf{p}}_{s_i}, \hat{\mathbf{y}}_{s_i}\right), \tag{4}$$

where $\lambda_1$ and $\lambda_2$ are hyperparameters balancing each term.

Besides, since $\mathbf{y}_s$ and $\bar{\mathbf{y}}_{s_i}$ are scribble annotations, we apply the *partial cross-entropy* (pCE) function [38] $L_{pce}$, which calculates the loss only for annotated pixels as the supervised loss, following [7, 8, 38]. Formally, let $\mathbf{m}$ and $\mathbf{n}$ be the prediction and the scribble annotation, and $L_{pce}(\mathbf{m}, \mathbf{n})$ is defined as

$$L_{pce}(\mathbf{m}, \mathbf{n}) = -\sum_{j \in J} \sum_{k \in K} \mathbf{n}^{jk} \log \mathbf{m}^{jk}, \tag{5}$$

in which $J$ is the set of pixels with scribble annotation, $K$ is the number of classification categories. $\mathbf{m}^{jk}$ indicates the predicted value of $k$-th channel for the $j$-th pixel in $\mathbf{m}$, and $\mathbf{n}^{jk}$ is the corresponding ground truth of $k$-th channel for the $j$-th pixel annotation in $\mathbf{n}$.

Lastly, we use the *cross-entropy* (CE) loss $L_{ce}$ as the unsupervised loss. Thus, Eq. (4) is re-written as

$$L_i = \underbrace{L_{pce}^{unmix}(\mathbf{p}_{s_i}, \mathbf{y}_s) + \lambda_1 L_{pce}^{mix}(\bar{\mathbf{p}}_{s_i}, \bar{\mathbf{y}}_{s_i})}_{\text{sup}} + \underbrace{\lambda_2 L_{ce}^{mix}(\bar{\mathbf{p}}_{s_i}.\hat{\mathbf{y}}_{s_i})}_{\text{unsup}}, \tag{6}$$

where the superscript *unmix* denotes that labels for calculation are original and without the mix augmentation. The superscript *mix* indicates that labels and pseudo-labels for calculation are generated from the mix augmentation.

## 4 Experiments on Semi-Supervised Segmentation

### 4.1 Data and Evaluation Metric

**ACDC dataset** [59] consists of 200 MRI volumes from 100 patients, and each volume manually delineates the ground truth for the left ventricle (LV), the right ventricle (RV), and the myocardium (Myo). The original volume sizes are $(154 - 428) \times (154 - 512) \times (6 - 18)$ pixels. We resized all the volumes to $256 \times 256 \times 16$ pixels and normalized the intensities as zero mean and unit variance. We performed *4-fold* cross-validation. We validated our method under the 16/150 partition protocol. In each fold, we sampled 16 volumes among 150 as the labeled data, and the remaining ones were treated as unlabeled data.

**Hippocampus dataset** was collected by The Medical Segmentation Decathlon[2], is comprised of 390 MRI volumes of the hippocampus. We utilized the training set (260 volumes) for validation, which contains the corresponding ground truth of the anterior and posterior regions of the hippocampus. Volume sizes are $(31 - 43) \times (40 - 59) \times (24 - 47)$ pixels. We resized all the volumes to $32 \times 48 \times 32$ pixels. With this dataset, we challenged a more tough problem where only one labeled sample is available for training, *i.e., one-shot setting*. We conducted *4-fold* cross-validation, sampled 1 volume among 195 cases as the labeled data in each fold, and treated the rest as unlabeled data.

**Evaluation metric.** Dice score and 95% Hausdorff Distance (95HD) were used to measure the volume overlap rate and the surface distance.

[2] http://medicaldecathlon.com/

**Table 1. Comparison with semi-supervised state-of-the-arts** on ACDC dataset under 16/150 partition protocol. We report the average (standard deviation) results based on *4-fold* cross-validation. †: method with ensemble strategy.

| method | RV | | Myo | | LV | | avg | |
|---|---|---|---|---|---|---|---|---|
| | Dice | 95HD | Dice | 95HD | Dice | 95HD | Dice | 95HD |
| upper bound | 81.6 (2.8) | 4.2 (2.3) | 79.5 (1.6) | 2.0 (0.3) | 89.6 (1.7) | 2.2 (0.6) | 83.6 | 2.8 |
| baseline | 58.9 (2.3) | 28.7 (8.3) | 56.1 (3.3) | 17.0 (3.1) | 70.4 (2.5) | 12.1 (5.3) | 61.8 | 19.3 |
| MT [4] | 58.1 (3.1) | 27.2 (9.3) | 58.0 (3.8) | 14.8 (1.4) | 70.5 (4.0) | 7.8 (3.6) | 62.2 | 16.6 |
| UA-MT [14] | 54.5 (7.5) | 35.4 (6.3) | 58.6 (3.2) | 17.9 (1.9) | 72.1 (3.1) | 10.3 (2.2) | 61.7 | 21.2 |
| CutMix-Seg [9] | 57.4 (2.7) | 36.1 (5.0) | 59.3 (4.2) | 18.8 (4.3) | 71.8 (2.1) | 14.2 (9.9) | 62.8 | 23.0 |
| STS-MT [28] | 57.1 (4.1) | 33.0 (5.4) | 60.1 (3.3) | 13.5 (2.5) | 72.0 (2.7) | 9.2 (3.1) | 63.1 | 18.6 |
| CPS [10] | 74.6 (3.2) | 7.0 (2.1) | 72.5 (2.0) | 5.0 (1.4) | 84.8 (1.2) | 5.5 (1.5) | 77.3 | 5.9 |
| UMCT [18] | 58.2 (2.8) | 29.1 (4.7) | 60.4 (2.9) | 16.4 (5.4) | 74.6 (2.3) | 11.1 (5.7) | 64.4 | 18.9 |
| UMCT† [18] | 61.9 (2.0) | 21.9 (4.6) | 63.2 (3.5) | 11.3 (5.4) | 78.3 (1.1) | 7.9 (5.0) | 67.8 | 13.7 |
| TriMix | 73.9 (3.5) | 7.9 (2.4) | 72.8 (1.7) | 4.3 (1.1) | 85.8 (1.7) | 4.7 (1.3) | 77.5 | 5.6 |
| TriMix† | **74.8 (3.6)** | **6.4 (2.0)** | **73.7 (1.9)** | **3.9 (1.1)** | **86.3 (1.7)** | **3.9 (1.2)** | **78.3** | **4.7** |

### 4.2 Experimental Setup

**Implementation details.** We adopted V-Net [58] as the backbone architecture. To fit the volumetric data, we extended CutMix [46] to 3D and set the cropped volume ratio to 0.2. We empirically set $\lambda$ to 0.5 in Eq. (3). We trained TriMix 300 epochs using SGD with a weight decay of 0.0001 and a momentum of 0.9. The initial learning rate was set to 0.01 and was divided by 10 every 100 epochs. At each training iteration, 4 labeled and 4 unlabeled samples were fetched for the ACDC dataset, and 1 labeled and 4 unlabeled samples were fetched for the Hippocampus dataset.

**Baseline and upper bound.** We provided the baseline and upper bound settings for reference. We trained the backbone V-Net only with the partitioned labeled data and treated the result as the baseline setting. Besides, we regraded the result trained with the complete labeled data as the upper bound accuracy.

**Mainstream approaches.** We implemented several SSL algorithms: Mean Teacher (MT) [4], Uncertainty-Aware Mean Teacher (UA-MT) [14], CutMix-Seg [9], Spatial-Temporal Smoothing Mean Teacher (STS-MT) [28], Uncertainty-Aware Multi-View Co-Training (UMCT) [18], and Cross Pseudo Supervision (CPS) [10], and compared TriMix to them. CutMix-Seg and CPS were incorporated with the 3D CutMix augmentation. UMCT was trained with three different views. We will report the student model results for MT, UA-MT, STS-MT, and CutMix-Seg. Since there is more than one trainable model within CPS and UMCT, we will report their average result among the trained models and the ensembled result for UMCT, the same as TriMix.

### 4.3 Experiment Results

**Improvement over the baseline.** We investigated TriMix's effectiveness in exploiting the unlabeled data. As illustrated in Table 1 and Table 2, we note that TriMix significantly improve the baseline. Specifically, it gains +15.7% in

**Table 2. Comparison with semi-supervised state-of-the-arts** on Hippocampus dataset with one-shot setting. We report the average (standard deviation) results based on *4-fold* cross-validation. †: method with ensemble strategy.

| method | anterior | | posterior | | avg | |
|---|---|---|---|---|---|---|
| | Dice | 95HD | Dice | 95HD | Dice | 95HD |
| upper bound | 84.4 (0.7) | 1.5 (0.1) | 82.6 (0.8) | 1.4 (0.1) | 83.5 | 1.5 |
| baseline | 12.9 (2.8) | 9.9 (1.7) | 14.7 (5.3) | 9.9 (1.5) | 13.8 | 9.9 |
| MT [4] | 25.2 (5.9) | 9.5 (1.2) | 29.2 (7.2) | 10.2 (1.6) | 27.2 | 9.9 |
| UA-MT [14] | 23.3 (2.4) | 8.5 (0.6) | 34.7 (9.7) | 9.3 (0.6) | 29.0 | 8.9 |
| CutMix-Seg [9] | 29.7 (5.6) | 7.5 (1.1) | 41.5 (10.8) | 8.9 (0.7) | 35.6 | 8.2 |
| STS-MT [28] | 26.1 (2.5) | 9.5 (1.1) | 31.3 (9.5) | 10.7 (1.3) | 28.7 | 10.1 |
| CPS [10] | 55.1 (4.6) | 5.9 (0.8) | 56.8 (2.7) | 4.5 (0.1) | 56.0 | 5.2 |
| UMCT [18] | 30.3 (8.9) | 6.9 (0.6) | 26.0 (9.8) | 5.4 (1.7) | 28.2 | 6.2 |
| UMCT† [18] | 35.3 (12.0) | 3.9 (1.0) | 27.6 (11.2) | 3.6 (1.2) | 31.5 | 3.8 |
| TriMix | 70.0 (3.4) | 3.0 (0.3) | 67.0 (2.1) | 3.2 (0.4) | 68.5 | 3.1 |
| TriMix† | **70.5 (3.6)** | **2.9 (0.4)** | **68.0 (1.7)** | **3.0 (0.3)** | **69.2** | **3.0** |

Dice and -13.7 in 95HD on the ACDC dataset and +54.7% in Dice and -6.8 in 95HD on the Hippocampus dataset, demonstrating that TriMix can effectively mine informative information from the unlabeled data to improve generalization.

**Comparison with SOTAs.** For the ACDC dataset under 16/150 partition protocol (see Table 1), CutMix-Seg achieves better average results than MT and confirms its effectiveness with strong input perturbation. STS-MT employs the spatial-temporal smoothing mechanism and outperforms CutMix-Seg. UMCT is in a co-training style and takes advantage of multi-view information. It brings higher accuracy than STS-MT but can not achieve the performance of CPS. TriMix obtains the best results among the methods. For the Hippocampus dataset with the one-shot setting (see Table 2), the existing SSL methods generally improve the baseline, verifying how effectively they exploit the unlabeled data. TriMix greatly outperforms the other methods, producing meaningful accuracy. Notably, TriMix surpasses the second-best method CPS by +12.5% in Dice and -2.1 in 95HD. Validation of these two datasets reveals that TriMix is competitive with SOTAs under typical partition protocols and has strong potential for learning from extremely scarce labeled data.

### 4.4 Empirical Study and Analysis

**Pseudo supervision consistency *vs.* probability consistency.** We compared the pseudo supervision consistency (denoted by $L^s_{unsup}$) and probability consistency (denoted by $L^p_{unsup}$) on the ACDC and Hippocampus datasets under different partition protocols. Results are shown in Fig. 3. Overall, TriMix incorporated with $L^s_{unsup}$ outperforms TriMix with $L^p_{unsup}$ across all the partition protocols on the two datasets. Especially under the one-shot setting on the Hippocampus dataset, $L^s_{unsup}$ surpasses $L^p_{unsup}$ by +54.2% in Dice and -5.9 in 95HD, indicating that a one-hot label map plays a more crucial role than a probability map as the expanded ground truth to supervise the other models within the

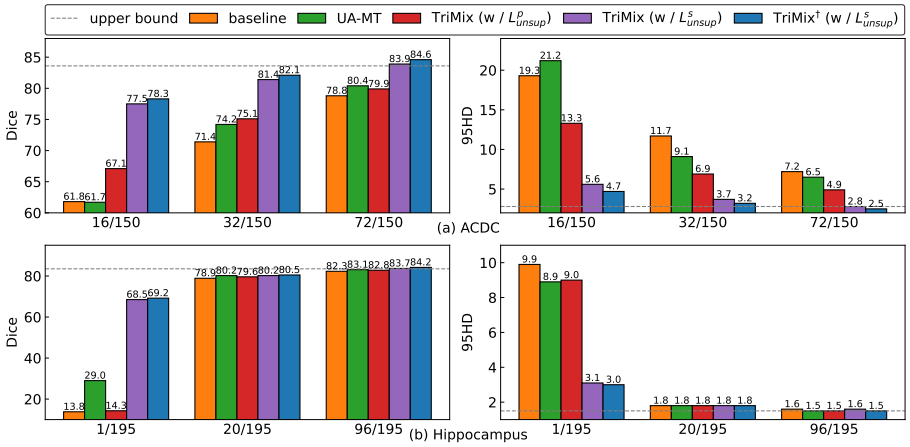


**Fig. 3. Empirical study** on different types of consistency regularization and various partition protocols with ACDC and Hippocampus datasets. $L^s_{unsup}$: an unsupervised loss that compares pseudo supervision consistency. $L^p_{unsup}$: an unsupervised loss that calculates probability consistency. †: method with ensemble strategy.

framework TriMix. Previous works [5, 8, 10] have reported similar observations. Using hard pseudo-labels encourages models to be low-entropy/high-confidence on data and is closely related to entropy minimization [60]. Based on this ablation, we utilize the pseudo supervision consistency as the *default setting* for TriMix in semi- and scribble-supervised segmentation.

**Robustness to different partition protocols.** We studied TriMix's robustness to various partition protocols on the ACDC and Hippocampus datasets. As shown in Fig. 3, TriMix consistently promotes the baseline and outperforms UA-MT across all the partition protocols, demonstrating the robustness and effectiveness of our method under different data settings. Moreover, TriMix surpasses the upper bound accuracy under the 72/150 partition protocol on the ACDC dataset and the 96/195 partition protocol on the Hippocampus dataset, revealing that TriMix can greatly reduce dependence on the labeled data.

**Relations to existing methods.** Among the semi-supervised methods for comparison, UMCT and CPS are the two most related methods to TriMix. UMCT is a co-training-based strategy to introduce view differences. Thus, TriMix resembles UMCT in some sense as both methods follow the spirit of multi-model joint training and encourage consistency among models. However, TriMix adopts a stricter perturbation than UMCT. Moreover, CPS can be regarded as a downgraded version of TriMix, in which two perturbed networks are trained to generate hard pseudo-labels to supervise each other. TriMix outperforms UMCT and CPS on the ACDC and Hippocampus datasets, demonstrating the superiority of our strategy, where consistency regularization under a more challenging perturbation is adopted in tri-training.

# 5 Experiments on Scribble-Supervised Segmentation

## 5.1 Data and Evaluation Metric

**ACDC dataset** [59] introduced in Section 4.1 was reused in this task, but with corresponding scribble annotations [6]. We resized all slices to the size of 256×256 pixels and normalized their intensity to [0,1], identical to the work [8].

**MSCMRseg dataset** [61] comprises of LGE-MRI images from 45 patients. We utilized the scribble annotations of LV, Myo, and RV released from [7] and used the same data partition setting as theirs: 25 images for training, 5 for validation, and 15 for testing. For data prepossessing, we re-sampled all images to the resolution of 1.37×1.37 mm, cropped or padded images to the size of 212×212 pixels, and normalized each image to zero and unit variance.

**Evaluation metric.** Dice score and 95HD were utilized.

## 5.2 Experimental Setup

**Implementation details.** We adopted the 2D U-Net architecture [62] as the backbone for all experiments in this task. The cropped area ratio was set to 0.2 when performing the CutMix augmentation. $\lambda_1$ and $\lambda_2$ in Eq. (6) were empirically set to 1. For the ACDC dataset, we used almost the same settings as in [8]. Specifically, we used SGD (weight decay = 0.0001, momentum = 0.9) to optimize TriMix for a total of 60000 iterations under a poly learning rate with an initial value of 0.03. The batch size was set to 12. We performed *5-fold* cross-validation. For the MSCMRseg dataset, we followed [7] to train TriMix 1000 epochs with the Adam optimizer and a fixed learning rate of 0.0001. We conducted *5 runs* with seeds 1, 2, 3, 4 and 5.

**Baseline and upper bound.** 2D U-Net trained with scribble annotations using the pCE loss [38] was regarded as the baseline setting. Furthermore, the upper bound accuracy was obtained using entirely dense annotations.

**Mainstream approaches.** We compared TriMix with several methods, including training with pseudo-labels generated by Random Walks (RW) [33], Scribble2Lables (S2L) [19], Uncertainty-Aware Self-Ensembling and Transformation Consistency Model (USTM) [39], Entropy Minimization (EM) [60], Mumford–Shah Loss (MLoss) [63], Regularized Loss (RLoss) [37], Dynamically Mixed Pseudo Labels Supervision (simply abbreviated to DMPLS in this paper) [8], CycleMix [7], and Shape-Constrained Positive-Unlabeled Learning (ShapePU) [20].

## 5.3 Experiment Results

**Improvement over baseline.** As shown in Table 3 and Table 4, TriMix significantly improves the baseline on the ACDC and MSCMRseg datasets, gaining +20.2% and +49.6% Dice scores, respectively, which proves that TriMix can learn good representations from sparse scribble annotations.

**Comparison with SOTAs.** For the ACDC dataset (see Table 3), TriMix achieves the highest average accuracy in Dice and 95HD among all scribble-supervised methods and reaches the closest result to the upper bound accuracy.

**Table 3. Comparison with scribble-supervised state-of-the-arts** on ACDC dataset. Other average (standard deviation) results are from [8]. Ours are based on *5-fold* cross-validation. †: method with ensemble strategy.

| method | RV | | Myo | | LV | | avg | |
|---|---|---|---|---|---|---|---|---|
| | Dice | 95HD | Dice | 95HD | Dice | 95HD | Dice | 95HD |
| upper bound | 88.2 (9.5) | 6.9 (10.8) | 88.3 (4.2) | 5.9 (15.2) | 93.0 (7.4) | 8.1 (20.9) | 89.8 | 7.0 |
| baseline | 62.5 (16.0) | 187.2 (35.2) | 66.8 (9.5) | 165.1 (34.4) | 76.6 (15.6) | 167.7 (55.0) | 68.6 | 173.3 |
| RW [33] | 81.3 (11.3) | 11.1 (17.3) | 70.8 (6.6) | 9.8 (8.9) | 84.4 (9.1) | 9.2 (13.0) | 78.8 | 10.0 |
| USTM [39] | 81.5 (11.5) | 54.7 (65.7) | 75.6 (8.1) | 112.2 (54.1) | 78.5 (16.2) | 139.6 (57.7) | 78.6 | 102.2 |
| S2L [19] | 83.3 (10.3) | 14.6 (30.9) | 80.6 (6.9) | 37.1 (49.4) | 85.6 (12.1) | 65.2 (65.1) | 83.2 | 38.9 |
| MLoss [63] | 80.9 (9.3) | 17.1 (30.8) | 83.2 (5.5) | 28.2 (43.2) | 87.6 (9.3) | 37.9 (59.6) | 83.9 | 27.7 |
| EM [60] | 83.9 (10.8) | 25.7 (44.5) | 81.2 (6.2) | 47.4 (50.6) | 88.7 (9.9) | 43.8 (57.6) | 84.6 | 39.0 |
| RLoss [37] | 85.6 (10.1) | 7.9 (12.6) | 81.7 (5.4) | 6.0 (6.9) | 89.6 (8.6) | 7.0 (13.5) | 85.6 | 6.9 |
| DMPLS [8] | 86.1 (9.6) | **7.9 (12.5)** | 84.2 (5.4) | 9.7 (23.2) | 91.3 (8.2) | 12.1 (27.2) | 87.2 | 9.9 |
| TriMix | 87.7 (2.8) | 8.9 (4.6) | 86.4 (2.2) | 4.3 (1.6) | 92.3 (3.0) | 4.4 (1.9) | 88.8 | 5.9 |
| TriMix† | **88.3 (2.6)** | 8.2 (4.1) | **86.8 (2.2)** | **3.7 (1.5)** | **92.6 (2.7)** | **3.8 (1.8)** | **89.3** | **5.2** |

It is worth noting that TriMix obtains a gain of 1.6% in Dice over DMPLS and a reduction of 1.0 in 95HD than RLoss. For the MSCMRseg dataset (see Table 4), TriMix surpasses all mix augmentation-based schemes, *i.e.*, MixUp, CutOut, CutMix, PuzzleMix, CoMixUp, and CycleMix, as well as two SOTAs, *i.e.*, CycleMix, and ShapePU. TriMix outperforms CycleMix by +7.4% and ShapePU by +2.2% and even improves the upper bound accuracy by +11.9% in Dice. Evaluations of these two benchmarks reveal that TriMix shows stronger generalization learning from sparse annotations than SOTAs.

### 5.4 Empirical Study and Analysis

**Ablation on different loss combinations.** We investigated the effectiveness of different loss combinations on the accuracy, as illustrated in Fig. 4. Only leveraging the original scribble annotations, $L_{pce}^{unmix}$ brings the lower bound accuracy. $L_{pce}^{mix}$ contributes to the performance and boosts the lower bound by +2.8% in Dice, showing that mix augmentation aids in *increasing scribble annotations* and thus improves accuracy. $L_{ce}^{mix}$ contributes much more than $L_{pce}^{unmix}$ and improves the lower bound by +41.0% in Dice, revealing that *pseudo supervision is essential* for TriMix. Besides, combining all losses yields the highest accuracy.

**Relations to existing methods.** TriMix is related to DMPLS and CycleMix. Specifically, DMPLS utilizes co-labeled pseudo-labels from multiple diverse branches to supervise single-branch output based on consistency regularization. CycleMix employs mix augmentation to increase scribble annotations and imposes consistency under the input perturbation. TriMix seems to be at the *middle ground*. It imports mix augmentation similar to CycleMix and enforces the consistency among various outputs with pseudo-label supervision, resembling DMPLS. TriMix incorporates valid features beneficial for scribble-supervised segmentation and achieves the new SOTA performance on two public benchmarks, *i.e.*, the ACDC and MSCMRseg datasets.

**Table 4. Comparison with scribble-supervised state-of-the-arts** on MSCMRseg dataset. Other average (standard deviation) results in Dice score are from [7, 20]. Ours are based on *5 runs*. †: method with ensemble strategy.

| method | RV | Myo | LV | avg |
|---|---|---|---|---|
| upper bound | 68.9 (12.0) | 72.0 (7.5) | 85.7 (5.5) | 75.5 |
| baseline | 5.7 (2.2) | 58.3 (6.7) | 49.4 (8.2) | 37.8 |
| MixUp [44] | 37.8 (15.3) | 46.3 (14.7) | 61.0 (14.4) | 48.4 |
| CutOut [45] | 69.7 (14.9) | 64.1 (13.6) | 45.9 (7.7) | 59.9 |
| CutMix [46] | 76.1 (10.5) | 62.2 (12.1) | 57.8 (6.3) | 65.4 |
| PuzzleMix [50] | 2.8 (1.2) | 63.4 (8.4) | 6.1 (2.1) | 24.1 |
| CoMixUp [47] | 5.3 (2.2) | 34.3 (6.7) | 35.6 (7.5) | 25.1 |
| CycleMix [7] | 79.1 (7.2) | 73.9 (4.9) | 87.0 (6.1) | 80.0 |
| ShapePU [20] | 80.4 (12.3) | 83.2 (4.2) | 91.9 (2.9) | 85.2 |
| TriMix | 86.5 (0.6) | 83.6 (0.4) | 92.2 (0.3) | 87.4 |
| TriMix† | **87.7 (0.7)** | **84.7 (0.4)** | **93.0 (0.2)** | **88.5** |

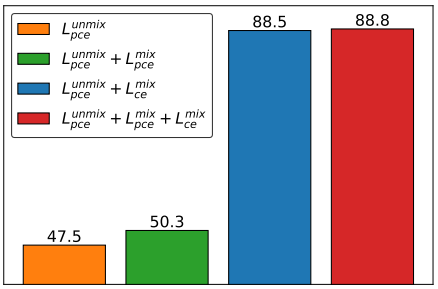


**Fig. 4. Ablation study on different loss combinations** on the ACDC dataset with scribble annotations using Dice score.

## 6 Discussion and Conclusion

This paper seeks to address semi- and scribble-supervised segmentation in a general way. We provide a hypothesis on a general learner learning from limited supervision: **(i)** it should output *more accurate predictions* and **(ii)** it should be trained with consistency regularization under a *more challenging perturbation*. We empirically verify the hypothesis with a simple framework. The method, called TriMix, purely imposes consistency on a tri-training framework under a stricter perturbation, *i.e.*, combining data augmentation and model diversity. Our method is competitive with task-specific mainstream methods. It shows strong potential training with extremely scarce labeled data and achieves new SOTA performance on two popular benchmarks when learning from sparse annotations. We also provide extra evaluations of our method in *appendix*.

Moreover, as suggested by [64], Deep Ensembles can provide a simple and scalable way for uncertainty estimation. TriMix maintains triple diverse networks, and such nature allows for its efficient uncertainty modeling. It is essential to estimate and quantify uncertainty for models learned from limited supervision, which is, however, rarely explored. It is also interesting to investigate whether TriMix can be applied to handle other types of imperfect annotations, *e.g.*, noise labels [2, 65]. In addition, TriMix's mechanism is similar to that of the method BYOL [66], which employs two networks and enforces representation consistency between them. TriMix may be applicable for self-supervised learning, but it needs further evaluation. Last but not least, similar to multi-view co-training [18], TriMix is inherently expensive in computation. To make TriMix more efficient, we may investigate strategies such as MIMO [67] for TriMix in the future. The above avenues are regarded as our follow-up works.

**Acknowledgement:** This work was supported by JSPS KAKENHI Grant Numbers 21K19898 and 17H00867 and JST CREST Grant Number JPMJCR20D5, Japan.

## References


1. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. *MedIA* **42** (2017)
2. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X.: Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *MedIA* **63** (2020)
3. Miyato, T., Maeda, S.i., Koyama, M., Ishii, S.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI* **41** (2018)
4. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: *NeurIPS*. (2017)
5. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In: *NeurIPS*. (2020)
6. Valvano, G., Leo, A., Tsaftaris, S.A.: Learning to segment from scribbles using multi-scale adversarial attention gates. *TMI* **40** (2021)
7. Zhang, K., Zhuang, X.: Cyclemix: A holistic strategy for medical image segmentation from scribble supervision. In: *CVPR*. (2022)
8. Luo, X., Hu, M., Liao, W., Zhai, S., Song, T., Wang, G., Zhang, S.: Scribble-supervised medical image segmentation via dual-branch network and dynamically mixed pseudo labels supervision. In: *MICCAI*. (2022)
9. French, G., Laine, S., Aila, T., Mackiewicz, M., Finlayson, G.: Semi-supervised semantic segmentation needs strong, varied perturbations. In: *BMVC*. (2020)
10. Chen, X., Yuan, Y., Zeng, G., Wang, J.: Semi-supervised semantic segmentation with cross pseudo supervision. In: *CVPR*. (2021)
11. Liu, Y., Tian, Y., Chen, Y., Liu, F., Belagiannis, V., Carneiro, G.: Perturbed and strict mean teachers for semi-supervised semantic segmentation. In: *CVPR*. (2022)
12. Ke, Z., Wang, D., Yan, Q., Ren, J., Lau, R.W.: Dual student: Breaking the limits of the teacher in semi-supervised learning. In: *ICCV*. (2019)
13. Ouali, Y., Hudelot, C., Tami, M.: Semi-supervised semantic segmentation with cross-consistency training. In: *CVPR*. (2020)
14. Yu, L., Wang, S., Li, X., Fu, C.W., Heng, P.A.: Uncertainty-aware self-ensembling model for semi-supervised 3D left atrium segmentation. In: *MICCAI*. (2019)
15. Wang, Y., Zhang, Y., Tian, J., Zhong, C., Shi, Z., Zhang, Y., He, Z.: Double-uncertainty weighted method for semi-supervised learning. In: *MICCAI*. (2020)
16. Luo, X., Chen, J., Song, T., Wang, G.: Semi-supervised medical image segmentation through dual-task consistency. In: *AAAI*. (2021)
17. Wu, Y., Xu, M., Ge, Z., Cai, J., Zhang, L.: Semi-supervised left atrium segmentation with mutual consistency training. In: *MICCAI*. (2021)
18. Xia, Y., Yang, D., Yu, Z., Liu, F., Cai, J., Yu, L., Zhu, Z., Xu, D., Yuille, A., Roth, H.: Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation. *MedIA* **65** (2020)
19. Lee, H., Jeong, W.K.: Scribble2label: Scribble-supervised cell segmentation via self-generating pseudo-labels with consistency. In: *MICCAI*. (2020)
20. Zhang, K., Zhuang, X.: Shapepu: A new pu learning framework regularized by global consistency for scribble supervised cardiac segmentation. In: *MICCAI*. (2022)

21. Zhou, Z.H., Li, M.: Tri-training: Exploiting unlabeled data using three classifiers. *TKDE* **17** (2005)
22. Zhao, A., Balakrishnan, G., Durand, F., Guttag, J.V., Dalca, A.V.: Data augmentation using learned transformations for one-shot medical image segmentation. In: *CVPR*. (2019)
23. Wang, S., Cao, S., Wei, D., Wang, R., Ma, K., Wang, L., Meng, D., Zheng, Y.: Lt-net: Label transfer by learning reversible voxel-wise correspondence for one-shot medical image segmentation. In: *CVPR*. (2020)
24. Tomar, D., Bozorgtabar, B., Lortkipanidze, M., Vray, G., Rad, M.S., Thiran, J.P.: Self-supervised generative style transfer for one-shot medical image segmentation. In: *WACV*. (2022)
25. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, *ICML*. (2013)
26. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: *IJCNN*. (2020)
27. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. In: *NeurIPS*. (2019)
28. Huang, T., Sun, Y., Wang, X., Yao, H., Zhang, C.: Spatial ensemble: a novel model smoothing mechanism for student-teacher framework. In: *NeurIPS*. (2021)
29. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: *ICLR*. (2017)
30. Li, S., Zhang, C., He, X.: Shape-aware semi-supervised 3d semantic segmentation for medical images. In: *MICCAI*. (2020)
31. Hang, W., Feng, W., Liang, S., Yu, L., Wang, Q., Choi, K.S., Qin, J.: Local and global structure-aware entropy regularized mean teacher model for 3d left atrium segmentation. In: *MICCAI*. (2020)
32. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *TPAMI* **23** (2001)
33. Grady, L.: Random walks for image segmentation. *TPAMI* **28** (2006)
34. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: *CVPR*. (2016)
35. Bai, W., Suzuki, H., Qin, C., Tarroni, G., Oktay, O., Matthews, P.M., Rueckert, D.: Recurrent neural networks for aortic image sequence segmentation with sparse annotations. In: *MICCAI*. (2018)
36. Ji, Z., Shen, Y., Ma, C., Gao, M.: Scribble-based hierarchical weakly supervised learning for brain tumor segmentation. In: *MICCAI*. (2019)
37. Tang, M., Perazzi, F., Djelouah, A., Ben Ayed, I., Schroers, C., Boykov, Y.: On regularized losses for weakly-supervised cnn segmentation. In: *ECCV*. (2018)
38. Tang, M., Djelouah, A., Perazzi, F., Boykov, Y., Schroers, C.: Normalized cut loss for weakly-supervised cnn segmentation. In: *CVPR*. (2018)
39. Liu, X., Yuan, Q., Gao, Y., He, K., Wang, S., Tang, X., Tang, J., Shen, D.: Weakly supervised segmentation of covid19 infection with scribble annotation on ct images. *PR* **122** (2022)
40. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: *CVPR*. (2019)
41. Hataya, R., Zdenek, J., Yoshizoe, K., Nakayama, H.: Faster autoaugment: Learning augmentation strategies using backpropagation. In: *ECCV*. (2020)
42. Lin, C., Guo, M., Li, C., Yuan, X., Wu, W., Yan, J., Lin, D., Ouyang, W.: Online hyper-parameter learning for auto-augmentation strategy. In: *ICCV*. (2019)

43. Tian, K., Lin, C., Sun, M., Zhou, L., Yan, J., Ouyang, W.: Improving auto-augment via augmentation-wise weight sharing. In: *NeurIPS*. (2020)
44. Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: *ICLR*. (2018)
45. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. *arXiv* (2017)
46. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: cutmix: Regularization strategy to train strong classifiers with localizable features. In: *ICCV*. (2019)
47. Kim, J., Choo, W., Jeong, H., Song, H.O.: Co-mixup: Saliency guided joint mixup with supermodular diversity. In: *ICLR*. (2021)
48. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: *ICML*. (2019)
49. Olsson, V., Tranheden, W., Pinto, J., Svensson, L.: Classmix: Segmentation-based data augmentation for semi-supervised learning. In: *WACV*. (2021)
50. Kim, J.H., Choo, W., Song, H.O.: Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In: *ICML*. (2020)
51. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: *COLT*. (1998)
52. Qiao, S., Shen, W., Zhang, Z., Wang, B., Yuille, A.: Deep co-training for semi-supervised image recognition. In: *ECCV*. (2018)
53. Peng, J., Estrada, G., Pedersoli, M., Desrosiers, C.: Deep co-training for semi-supervised image segmentation. *PR* **107** (2020)
54. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: *ICML*. (2017)
55. Chen, D.D., Wang, W., Gao, W., Zhou, Z.H.: Tri-net for semi-supervised deep learning. In: *IJCAI*. (2018)
56. Zhang, T., Yu, L., Hu, N., Lv, S., Gu, S.: Robust medical image segmentation from non-expert annotations with tri-network. In: *MICCAI*. (2020)
57. Yu, J., Yin, H., Gao, M., Xia, X., Zhang, X., Viet Hung, N.Q.: Socially-aware self-supervised tri-training for recommendation. In: *KDD*. (2021)
58. Milletari, F., Navab, N., Ahmadi, S.: V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In: *3DV*. (2016)
59. Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.A., Cetin, I., Lekadir, K., Camara, O., Gonzalez Ballester, M.A., Sanroma, G., Napel, S., Petersen, S., Tziritas, G., Grinias, E., Khened, M., Kollerathu, V.A., Krishnamurthi, G., Rohé, M.M., Pennec, X., Sermesant, M., Isensee, F., Jäger, P., Maier-Hein, K.H., Full, P.M., Wolf, I., Engelhardt, S., Baumgartner, C.F., Koch, L.M., Wolterink, J.M., Išgum, I., Jang, Y., Hong, Y., Patravali, J., Jain, S., Humbert, O., Jodoin, P.M.: Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved? *TMI* **37** (2018)
60. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: *NeurIPS*. (2004)
61. Zhuang, X.: Multivariate mixture model for myocardial segmentation combining multi-source images. *TPAMI* **41** (2018)
62. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: *MICCAI*. (2015)
63. Kim, B., Ye, J.C.: Mumford–shah loss functional for image segmentation with deep learning. *TIP* **29** (2019)
64. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: *NeurIPS*. (2017)

65. Karimi, D., Dou, H., Warfield, S.K., Gholipour, A.: Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *MedIA* **65** (2020)
66. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. In: *NeurIPS*. (2020)
67. Havasi, M., Jenatton, R., Fort, S., Liu, J.Z., Snoek, J., Lakshminarayanan, B., Dai, A.M., Tran, D.: Training independent subnetworks for robust prediction. In: *ICLR*. (2020)