Received: 5 July 2021 | Revised: 1 March 2022 | Accepted: 9 April 2022

IET Image Processing

DOI: 10.1049/ipr2.12511

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

ROAM: Random layer mixup for semi-supervised learning in medical images

Tariq Bdair[1] | Benedikt Wiestler[2] | Nassir Navab[1,3] | Shadi Albarqouni[1,4,5]

[1] Chair for Computer Aided Medical Procedures & Augmented Reality, Technical University of Munich, Munich, Germany

[2] Department of Neuroradiology, Technical University of Munich, Munich, Germany

[3] Whiting School of Engineering, Johns Hopkins University, Baltimore, MD, USA

[4] Helmholtz AI, Helmholtz Center Munich, Neuherberg, Germany

[5] Clinic for Diagnostic and Interventional Radiology, University Hospital Bonn, Venusberg-Campus 1, Bonn, Germany

Correspondence
Tariq Bdair, Chair for Computer Aided Medical Procedures & Augmented Reality, Technical University of Munich, 85748 Munich, Germany.
Email: t.bdair@tum.de

Funding information
Deutscher Akademischer Austauschdienst, Grant/Award Number: Tariq Bdair

Abstract

Medical image segmentation is one of the major challenges addressed by machine learning methods. However, these methods profoundly depend on a large amount of annotated data, which is expensive and time-consuming. Semi-supervised learning (SSL) approaches this by leveraging an abundant amount of unlabeled data. Recently, MixUp regularizer has been introduced to SSL methods by augmenting the model with new data points through linear interpolation at the input space. While this provides the model with new data, it is limited and may lead to inconsistent soft labels. It is argued that the linear interpolation at different representations provides the network with novel training signals and overcomes the inconsistency of the soft labels. This paper proposes ROAM as an SSL method that explores the manifold and performs linear interpolation on randomly selected layers to generate virtual data that has never been seen before, which encourages the network to be less confident for interpolated points. Hence it avoids overfitting, enhances the generalization, and shows less sensitivity to the domain shift. Extensive experiments are conducted on publicl datasets on whole-brain and lung segmentation. ROAM achieves state-of-the-art results in fully supervised (89.5%) and semi-supervised (87.0%) settings with relative improvements up to 2.40% and 16.50%, respectively.

1 | INTRODUCTION

Medical imaging is defined as the process of capturing the interior of a body for clinical use, medical intervention, as well as visual monitoring of the function of some organs or tissues [1]. Medical image annotation plays a fundamental role in the medical field [2] since it provides a tool to examine different diseases [3], quantify human organs [4], therapy planning [5], tumor development monitoring [6, 7], diagnostic aid systems [8], and intra-operative assistance [9]. Nevertheless, manual segmentation is a tedious task that requires highly experienced physicians [10] and is subject to intra-/inter-observer variability [11]. That led to a great interest in automated segmentation methods estimated at 70% of international image analysis challenges in the medical domain [12]. Recently, deep learning-based methods have achieved state-of-the-art performance in medical image segmentation [13–15], and shown their applicability to a wide range of datasets without requiring human expert [16]. However, one major drawback of this approach is the necessity for a huge amount of annotated data which is oftentimes not available in medical images.

Fortunately, the semi-supervised learning (SSL) framework provides the tool to alleviate this problem by utilizing a huge amount of unlabeled data along with a few annotated ones in intelligent and efficient ways. Thus, SSL methods have proved their benefits to real cases which fit the nature of medical data, where the scarcity of labeled data is the main characteristic. While surveying SSL methods is out of the scope of this paper, we refer the reader to references [17, 18] for a useful introduction. In general, SSL methods can be grouped into four main categories; (i) generative model, (ii) graph-based methods, (iii) entropy minimization, and (iv) consistency regularization. Next, we introduce briefly these methods with the focus on SSL works on medical images.

Generative models have been extensively used in the past few years to estimate the density distribution of the data using the concept of adversarial learning [19]. Specifically, two networks were used in the training process, namely the generator and the discriminator networks. The goal of the generator is to produce fake data of parallel high quality to that of the original data, while the goal of the discriminator is to distinguish between the fake and the original data. This idea has been utilized by Zhang et al. [20] for gland image segmentation by encouraging the discriminator to distinguish between the segmentation results of unlabeled and labeled images while encouraging the segmenter (generator) to produce results fooling the discriminator. Nie et al. [21] utilized attention-based approach, based on the confidence map from the confidence network (discriminator), to include the unlabeled data in the adversarial training for pelvic organ segmentation. Chen et al. [22] encouraged the model to learn discriminative features for segmentation from unlabeled images, using an autoencoder trained to synthetic segmentation labels, to segment tumor and white matter hyperintensities in the brain. SCLLD [23] proposed GAN-based architecture consisting of two training phases to detect COVID-19 infection. First, the weights of generator and discriminator networks are initialized using the unlabeled data, then fine-tuned by exploiting the labeled ones. VTGAN [24], however, proposed a semi-supervised GAN-based method to synthesize retinal vascular structure angiograms from fundus images while detecting healthy and abnormal retina. Transformer-based discriminators take the original and generated images then produce feature maps used for disease classification. Major drawbacks of these approaches include the computation overhead and the complexity in the architecture. For instance, reference [23] involves two training stages, while VTGAN [24] consists of four networks; two generators and two transformer-based discriminators. In contrast, our method is easy to implement, consists of one backbone network, and requires no additional training phases.

Graph-based methods represent the data, both labeled and unlabeled, in a graph structure, where the nodes represent the data points, the edges represent the connectivity, and the weights represent the distance between the nodes. Graphs can be used to propagate the labels from the labeled data to the unlabeled ones based on the connectivity and similarity. Baur et al. [25] introduced this concept as a regularization term to the main objective function for MS lesion segmentation. The term is based on the Laplacian graphs and attempts to minimize the distance between similar unlabeled and labeled data points in the hidden space. Ganaye et al. [26] took the advantages of the invariant nature of the brain structure to build an adjacency graph of the brain structures acting as a constraint to refine the predicted segmentation of the unlabeled data. Graph convolutional networks (GCNs)-based approaches have been proposed to handle the unstructured format of some medical data. For instance, GKD [27] distilled the knowledge from teacher to a student model. The teacher graph injects the available information into soft pseudo labels. Then, the pseudo labels are used to train a student graph for Autism spectrum disorder or Alzheimer's disease prediction. RA-GCN [28], on the other hand, addressed the imbalanced class distribution in the medical data by representing each class by a graph-based neural network responsible for the weighting of class samples. The whole architecture, then, is trained in an adversarial manner such that the classifier adapts itself with the attention to rare cases. The previous methods have shown their benefits to unstructured data, yet, it suffers the burden of graph construction and weighing steps. Moreover, graph-based approaches are optimized on the unlabeled data, which results in a lack of scalability. In contrast, our method is optimized on the whole input space and relaxes the need for any previous steps.

Entropy minimization forces the decision boundary to pass through low-density regions to minimize the entropy of the predictions. One way to achieve this, in SSL setting, is to generate pseudo labels for the unlabeled data using a model trained on the labeled data. Next, the training process is repeated using both labeled and pseudo-labeled data [29]. This approach has been employed by Bai et al. [30] for cardiac image segmentation, where the pseudo labels were additionally fine-tuned using the conditional random field (CRF) method. Close to pseudo labeling is co-training [31] where confident predictions from separate models, trained using different views of the data, are utilized to enhance the training. Xia et al. [32] utilized co-training by enforcing multi-view consistency of the unlabeled data for pancreas and multi-organ segmentation. PLAT [33] exploited an adaptive threshold that avoids the noisy signals and generates more accurate pseudo labels to detect the cells in microscopic and stained histology images. Reference [34] proposed a pseudo labeling approach, namely self-loop uncertainty that exploited self-supervised learning sub-task that solves Jigsaw puzzles to mine the information from the unlabeled data to help in the training. While the FCN-based network is optimized to solve Jigsaw puzzles, it produces different segmentation predictions (corresponding to each stage). Then, these predictions are averaged and used as uncertainty estimation yielded by ensembling multiple models to improve the segmentation accuracy in stained tissue and skin lesion images. Our method is similar to the previous ones in the pseudo labeling step. However, we are different by two folds. First, the aforementioned methods generate pseudo labels for the unlabeled data only. Yet, our method, in addition to that, generates virtual data points and their corresponding pseudo labels from linear interpolation at a random layer of the input data. This process augments the model with novel training signals that have never been seen before, see Section 2.3 for more details. Second, the previous methods utilize different post-processing steps to enhance the quality of the pseudo labels, yet, none of them used a sharpening operation that pushes the pseudo labels into more confident regions, which was adopted by our method, check Section 2.3 and Figure 1 for more details.

Consistency regularization methods train the model to predict the same output for different perturbations or augmentations of the input data. Mean-Teacher [35], one of the most successful method of consistency regularization, has been employed by Cui et al. [36] for brain lesion segmentation. They simply introduce a segmentation consistency loss to minimize the discrepancy between the outputs of unlabeled data under different perturbations. A similar approach

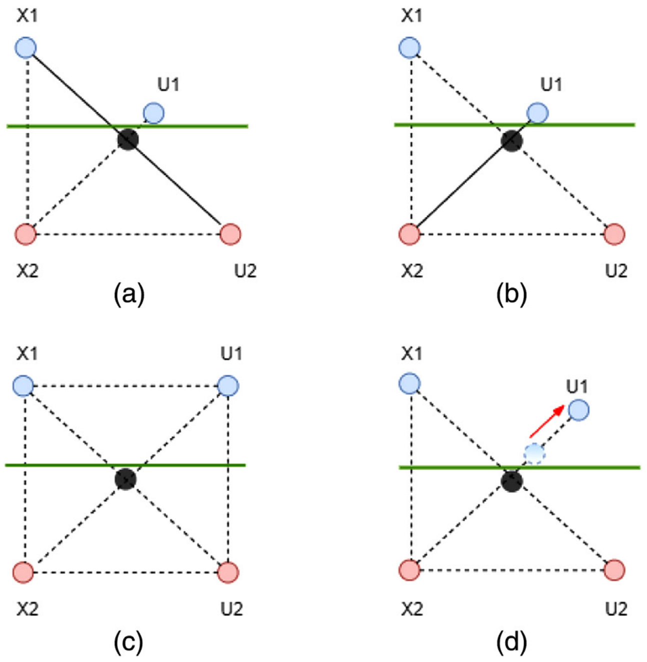


FIGURE 1 Illustrative example. (a, b) Input Mixup: Shows the inconsistency of the generated soft label of grey-dot resulted from two different linear interpolations of inputs. (c) Manifold Mixup: The learned hidden states are better organized in local regions leading to the consistency of the soft labels. (d) Sharpening operation (red arrow) pushes the soft label of to a more confident region

was utilized by Bortsova et al. [37] for chest x-ray images segmentation. Yu et al. [38] included the uncertainty information to enable the student model to learn from the reliable targets for left atrium segmentation. Li et al. [39] utilized transformation-consistency to enhance the regularization on the pixel level. Interesting results were demonstrated on skin lesion, optic disk, and liver segmentation. UDC-Net [40] forced the so-called dual consistency between the predictions of unlabeled images on one side and the predictions of its transformed version and auxiliary decoders on the other side. Further, the consistency is guided by uncertainty measures and applied for COVID-19 lesion segmentation in the CT scans. UATS [41] applied the consistency between the current prediction of unlabeled images and its ensemble predictions from previous epochs for prostate segmentation. Yet, Wang et al. [42], in addition to the consistency between different augmentations of the unlabeled images, forced consistency between the input images and their adversarial direction to classify breast cancer in ultrasound images and ophthalmic disease in the OCT scans. Except for UDC-Net, all the previous methods applied data augmentation at the input space to force the consistency loss. In contrast, our approach augments the images at the input and the hidden layers. Although UDC-Net proposed the perturbations at the features level, they introduced a sophisticated augmentation process consisting of seven decoders. In contrast, our method handles that by simply utilizing a linear interpolation. Moreover, UDC-Net used seven decoders fixed at one hidden space to create different variations. However, our method overcomes this limitation by randomly selecting the hidden spaces on which the augmentation is performed.

Modern regularization methods such as Input MixUp [43], and Manifold Mixup [44] have been recently introduced to avoid overfitting by encouraging the model to be less confident for interpolated data points at the input space or the latent space, respectively. Both methods have been successfully employed for fully supervised segmentation frameworks; for example, cardiac image segmentation [45], brain tumor segmentation [46], knee segmentation [47], and prostate cancer segmentation [48]. While the previous works have shown the effectiveness of MixUp over standard data augmentation methods in medical images, they depend heavily on fully labeled datasets, which usually are expensive and not available. Yet, this paper addresses the scarcity of the labeled data by proposing a SSL approach.

Recently, MixMatch [49], that inspired our work, introduced Input MixUp to the SSL paradigm achieving SOTA results in image classification. MixMatch augments the model with interpolated data between labeled and unlabeled images at the input space. While this approach is interesting and indeed provides the model with diverse data points, it is rather limited, and suffers from inconsistent soft labels for the interpolated data points. We argue that performing the mixup operation at *randomly* selected input and hidden representations of the labeled and the unlabeled data provides the network with novel representations and additional training signals that suit the complexity of medical image segmentation tasks. Moreover, it provides stable soft labels of the augmented samples. Our method takes the advantages of both MixMatch and Manifold Mixup to boost the performance of the model leading to better generalizability. Thus, our **contributions** can be listed as follows:

- Proposing *RandOm lAyer Mixup (ROAM)* that explores the manifold by randomly selecting a subset of input and hidden layers to perform a linear interpolation of labeled and unlabeled data points and generate virtual data that fits the complexity of medical imaging segmentation in both fully and semi-supervised settings.
- ROAM overcomes the limitations of the previous methods by encouraging the network to be less confident for interpolated data points and reducing overfitting and generalizing well to unseen data.
- Performing a comprehensive ablation study showing the importance of our design choices. Further, we discuss employing the Manifold Mixup with the presence of skip connections in U-Net-like architectures.
- Extensive experiments are performed, following the recommendations of Oliver et al. [50], to evaluate our method under the presence of domain shift, class mismatch, and different amounts of un-/labeled data.
- We empirically show the effectiveness of ROAM by demonstrating a SOTA performance in both supervised and semi-supervised settings in the whole-brain image segmentation and beating the baseline models in COVID-19 infection and lung segmentation.

- A unified architecture is utilized to implement different SSL methods for the sake of fair comparison. The code is made publicly available for benchmarking.

2 | METHODOLOGY

2.1 | SSL paradigm

In the SSL, we are given a set of labeled $\mathcal{S}_L = \{\mathcal{X}_L, \mathcal{Y}_L\}$ and unlabeled data $\mathcal{S}_U = \{\mathcal{X}_U\}$, where $\{\mathcal{X}_L, \mathcal{X}_U\} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_L, \boldsymbol{x}_{L+1}, \dots, \boldsymbol{x}_{L+U}\}$ are input images, $\boldsymbol{x} \in \mathbb{R}^{H \times W}$, and $\mathcal{Y}_L = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_L\}$ are the segmentation maps, $\boldsymbol{y} \in \mathbb{R}^{H \times W \times C}$, for C organs. Our goal is to build a model $\mathcal{F}(\boldsymbol{x}; \Theta)$ that takes input image $\boldsymbol{x}$ and outputs its segmentation map $\hat{\boldsymbol{y}}$. To leverage both labeled and unlabeled data in SSL paradigm, the objective function takes the form

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Supervised}} + \beta \mathcal{L}_{\text{Unsupervised}}, \tag{1}$$

where $\mathcal{L}_{\text{Supervised}}$ denotes the supervised loss and trained using labeled data $\mathcal{S}_L$, $L_{\text{Unsupervised}}$ denotes the unsupervised loss and trained on the unlabeled data $\mathcal{S}_U$, and β is a weighing factor that controls the contribution of the unsupervised loss.

The unsupervised loss can have different forms depending on the employed SSL aforementioned approaches. Here, we will focus on the consistency regularization approach, where its goal is to minimize the distance between the feature representations of the input data point x and its perturbed version $\hat{x}$. Formally, $L_{\text{Unsupervised}} = d(f_\theta(x), f_\theta(\hat{x}))$, where $d(\cdot, \cdot)$ is a distance metric.

2.2 | Preliminaries

Input Mixup [43] is a simple data augmentation method that generates new data points (x_k, y_k) through a linear interpolation between a pair of training examples (x_i, y_i) and (x_j, y_j),

$$x_k = \lambda x_i + (1 - \lambda) x_j, \tag{2}$$

$$y_k = \lambda y_i + (1 - \lambda) y_j, \tag{3}$$

where $\lambda \in [0, 1]$. Mixup is considered as a type of data augmentation where the newly generated data points extend the training dataset following the cluster and manifold assumptions [51] that linear interpolations of input examples should lead to linear interpolations of the corresponding labels.

One major drawback of this approach is that the interpolations between two samples may lead to inconsistent soft labels at interpolated points. Thus Input Mixup can suffer from underfitting and high loss. This can be better understood by examples. Figure 1 shows an illustrative example, where the red and the blue circles represent two classes. In Figure 1a, the grey-dot is generated by the linear combinations of a blue labeled example (X1) and a red unlabeled example (U2). Since the grey-dot is located in the middle distance between the two classes, based on the mixing factor λ, the generated soft label has an equal probability of blue and red classes (50% each). In contrast, in Figure 1b, the same data point (grey-dot) has been generated from a combination of X2 (red class) and U1 (blue class) with probability of 90% of being blue and 10% of being red, as it is located closer to U1, which leads to the inconsistency of the generated soft labels between the different scenarios.

Manifold Mixup [44], on the other hand, overcomes the above limitations by performing the mixup operation at the hidden layers. Thus, training is carried out on the convex combinations of the hidden representations of data samples. The learned representations lead to better organization of the hidden state for each class, where it is more concentrated and organized. As a result, the inconsistency of soft labels at interpolated points can be avoided. This can be shown in Figure 1c, where the generated soft label of grey-dot is consistent, with an equal probability of each class, regardless of the interpolated data points (X1 and U2 or X2 and U1).

2.3 | ROAM

The core components of our method are (a) *Pseudo labeling*: Given a pre-trained model for a few epochs on labeled data, the initial labels for the unlabeled batch were produced, then refined by applying a sharpening operation. (b) *ROAM*: The labeled batch and the unlabeled batch were concatenated, then passed to the network as normal. Then, a mixup operation is applied at a random layer, where the paired examples are randomly selected. At the same time, a mixup operation is applied to the corresponding labels. Finally, the process is continued from that layer to the output layer. In the following sections, we illustrate our methodology in detail, while the entire framework and the algorithm are shown in Figure 2 and Algorithm 1, respectively.

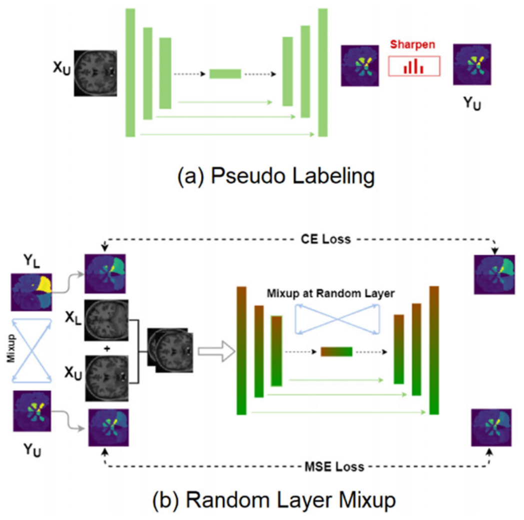
2.3.1 | Pseudo labels

First, the unlabeled data along with the labeled set are leveraged using two steps: (i) sharpening the initial predictions for unlabeled data to minimize its entropy following reference [49], and (ii) mixup the labeled and unlabeled data at random layers following reference [44]. The unlabeled data are first fed to the model outputting the initial predictions

$$\hat{\boldsymbol{y}}_i = \mathcal{F}(\boldsymbol{x}_i; \Theta); \quad \text{where} \quad \boldsymbol{x}_i \in \mathcal{X}_U, \tag{4}$$

before being post-processed by a sharpening operation, parameterized with T, which is highly inspired by the entropy minimization literature [49, 52]. The pseudo label set is then defined as $\tilde{\mathcal{Y}}_U = \{\tilde{\boldsymbol{y}}_i, \dots, \tilde{\boldsymbol{y}}_U\}$, where

$$\tilde{\boldsymbol{y}}_i = \texttt{Sharpening}(\hat{\boldsymbol{y}}_i, T)_j := \hat{\boldsymbol{y}}_{ij}^{\frac{1}{T}} \Big/ \sum_{j=1}^{C} \hat{\boldsymbol{y}}_{ij}^{\frac{1}{T}}, \tag{5}$$



(a) Pseudo Labeling

(b) Random Layer Mixup

FIGURE 2 Illustration of ROAM. (a) First, initial labels for the unlabeled batch are produced from a pre-trained model, then, a sharpening step is applied to fine-tune the labels. (b) Second, the labeled and unlabeled batches are fed to the network, and mixed at a random layer. Both models in (a) and (b) are the same, yet we freeze the parameters in step (a)

ALGORITHM 1 ROAM: Random Layer MixUp for SSL

```
Require: pre-trained model $\mathcal{F}(\cdot;\Theta^{(0)})$, labeled dataset $\mathcal{S}_L$, unlabeled dataset
  $\mathcal{S}_U$, batch size $B$, number of iteration $K$, The hyper-parameters $\{T,\alpha,\beta\}$
  Initialize: $k \longleftarrow 0, \Theta \longleftarrow \Theta^{(0)}$
while $k \leq K$do
    $\mathcal{B}_L \sim (\mathcal{X}_L, \mathcal{Y}_L)$;   $\mathcal{B}_U \sim \mathcal{X}_U$ //sample labeled and unlabeled batches
    $\hat{y}_i = \mathcal{F}(x_i;\Theta)$; $x_i \in \mathcal{B}_U$ //initial labels for$\mathcal{X}_U$; Equation (4)
    $\tilde{y}_i = \texttt{Sharpening}(\hat{y}_i, T)$ //pseudo labels; Equation (5)
    $\mathcal{X} = \{\mathcal{X}_L, \mathcal{X}_U\}, \mathcal{Y} = \{\mathcal{Y}_L, \tilde{\mathcal{Y}}_U\}$ //concatenate both batches, $\tilde{\mathcal{Y}}_U$ from
    Equation (5)
    $\kappa \longleftarrow$ randomly select layer
    $\mathcal{H} = \mathcal{F}_{\notin}(\mathcal{X})$ //pass the data to the network, and extract$\mathcal{H}$; Equation (6)
    $\tilde{\mathcal{H}}, \tilde{\mathcal{Y}} = \texttt{Permute}(\mathcal{H}, \mathcal{Y})$ //randomly shuffle the data
    $\mathcal{H}', \mathcal{Y}' = \texttt{Mixup}(\alpha, \mathcal{H}, \mathcal{Y}, \tilde{\mathcal{H}}, \tilde{\mathcal{Y}})$ //perform mixup operation; Eqs. (7,8)
    $\mathcal{P} \longleftarrow$ resume passing $\mathcal{H}'$ from layer $\kappa$ to the output layer
    $\mathcal{P}_{\mathcal{L}}, \mathcal{P}_{\mathcal{U}} = \texttt{Split}(\mathcal{P})$; $\mathcal{Y}'_{\mathcal{L}}, \mathcal{Y}'_{\mathcal{U}} = \texttt{Split}(\mathcal{Y}')$ //split the predictions and
    labels
    $\Theta \longleftarrow \arg\min_{\Theta} \mathcal{L}_{CE}(\mathcal{Y}'_L, \mathcal{P}_L) + \beta \mathcal{L}_{MSE}(\mathcal{Y}'_U, \mathcal{P}_U)$ //calculate the loss;
    Equation (9)
end while
```

where $\hat{y}_j$ is given by Equation (4), $j \in C$, and C is the total number of classes. Note that as $T \rightarrow 0, y_j$ approaches one-hot encoding. Applying the sharpening operation to the initial labels produces more stable predictions through pushing the labels away from the decision boundaries, specifically, to more confident regions for each class by minimizing its entropy. This effect can be easily seen in Figure 1d where the unlabeled data point U1 is moved closer to the right distribution.

2.3.2 | Random layer mixup

Given the unlabeled data $\mathcal{X}_U$ and its pseudo labels $\tilde{\mathcal{Y}}_U$, along with the labeled data $\mathcal{X}_L$ and its one-hot encoding labels $\mathcal{Y}_L$, the two sets are concatenated as $\mathcal{X} = \{\mathcal{X}_L, \mathcal{X}_U\}, \mathcal{Y} = \{\mathcal{Y}_L, \tilde{\mathcal{Y}}_U\}$. To enable running the mixup operation at randomly selected latent space, we first define $(\mathcal{H}, \mathcal{Y})$, where

$$\mathcal{H} = \begin{cases} \mathcal{X}, & \kappa = 0 \\ \mathcal{F}_{\kappa}(\mathcal{X}), & \text{otherwise} \end{cases}, \tag{6}$$

where $\mathcal{F}_{\kappa}(\cdot)$ is the hidden representation of the input data at layer κ. Note that the input data is selected when $\kappa = 0$. To introduce a noisy interpolated data, a permuted version of the original data is created $\tilde{\mathcal{H}}, \tilde{\mathcal{Y}} = \texttt{Permute}(\mathcal{H}, \mathcal{Y})$, and fed to the $\texttt{MixUp}$ operation as

$$\mathcal{H}' = \lambda' \mathcal{H} + (1 - \lambda')\tilde{\mathcal{H}}, \tag{7}$$

$$\mathcal{Y}' = \lambda' \mathcal{Y} + (1 - \lambda')\tilde{\mathcal{Y}}, \tag{8}$$

where $\texttt{Permute}(.)$ randomly shuffles the data, $\mathcal{H}'$ and $\mathcal{Y}'$ are the interpolated mixed-up data, where the paired examples are selected randomly.

To favour the original data over the permuted one, λ' is set to $\max(\lambda, 1 - \lambda)$, where $\lambda \in [0, 1]$ is sampled from a $\text{Beta}(\alpha, \alpha)$ distribution with α as a hyper-parameter. Further, to keep the flow of the original data, we run some experiments without the mixup operation, and denoted as $\kappa = \Phi$. In practice, this can be achieved by setting κ and λ' to 0 and 1, respectively. To this end, the mixed-up data $\mathcal{H}'$ are fed to the model from layer κ along the way to the output layer at which the segmentation maps are predicted $\mathcal{P}$. Eventually, $\mathcal{P}$ is split back into labeled and unlabeled predictions $\mathcal{P} = \{\mathcal{P}_L, \mathcal{P}_U\}$, and similarly $\mathcal{Y}'$ into $\mathcal{Y}'_L$ and $\mathcal{Y}'_U$.

2.3.3 | Overall objective function

Our overall objective function is the sum of the cross entropy loss $\mathcal{L}_{\text{CE}}$ on the mixed-up labeled data, and the consistency mean squared loss $\mathcal{L}_{\text{MSE}}$ on the mixed-up unlabeled data,

$$\arg\min_{\Theta} \mathcal{L}_{\text{CE}}(\mathcal{Y}'_L, \mathcal{P}_L) + \beta \mathcal{L}_{\text{MSE}}(\mathcal{Y}'_U, \mathcal{P}_U), \tag{9}$$

where β is a hyper-parameter.

3 | EXPERIMENTS

Our experiments involve two parts; the whole-brain segmentation (Section 4) and lung segmentation (Section 5). First, a comparison with SSL methods for medical image segmentation is performed (Section 4.1), followed by a comparison with SOTA methods for whole-brain segmentation in a fully supervised setting (Section 4.4). Then, extensive experiments, following the recommendations of reference [50], are performed (Section 4.5). Further, the performance of ROAM is investigated in the presence of the domain shift (Section 4.5.1). In the second part, lung segmentation results are reported in semi and fully supervised fashions (Section 5.1). Then, ROAM is investigated in the presence of domain shift and classes mismatch (Section 5.2). Finally, the performance versus infection size is discussed (Section 5.3).

3.1 | Datasets

3.1.1 | Brain

For whole-brain segmentation, we opt for three publicly available datasets as follows: (i) MALC [53], which consists of 30 T1 MRI volumes, with manual segmentation for the whole brain which is provided by reference [54]. This dataset is divided into 15 training volumes and 15 testing volumes (∼2500 slices each). The training volumes further split into three labeled (∼500 slices), nine unlabeled volumes (∼1500 slices), and three validation volumes (∼500 slices). (ii) IBSR [55], which consists of 18 T1 MRI volumes (∼2000 slices). This dataset is provided with manual segmentation for the whole brain. (iii) CANDI [56], which consists of 13 T1 MRI volumes (∼1500 slices). The manual segmentation for whole brain for this dataset is provided by Neuromorphometrics, Inc. The labels for whole-brain segmentation include 27 classes (27 internal structures); Left Cortical WM, Left Cortical GM, Right Cortical WM, Right Cortical GM, Left Lateral Ventricle, Left Cerebellar WM, Left Cerebellar GM, Left Thalamus, Left Caudate, Left Putamen, Left Pallidum, 3rd Ventricle, 4th Ventricle, Brain Stem, Left Hippocampus, Left Amygdala, Left Ventral DC, Right Lateral Ventricle, Right Cerebellar WM, Right Cerebellar GM, Right Thalamus, Right Caudate, Right Putamen, Right Pallidum, Right Hippocampus, Right Amygdala, and Right Ventral DC.

3.1.2 | Lung

Two publicly available datasets for lung segmentation are used. (i) COVID-19-CT-Seg-Benchmark [57]: which consists of 20 CT volumes with the segmentation of three classes; right lung, left lung, and infection. The data is divided into 10 training volumes and 10 testing volumes (∼2000 slices each). The training data is further divided into two labeled volumes (∼300 slices), seven unlabeled volumes (∼1400 slices), and one validation volume (∼300 slices). (ii) MedSeg: which consists of 100 axial CT images (i.e. slices) from more than 40 patients with COVID-19. The images are divided into 80 training images and 20 validation images. The labels include three classes; ground-glass, consolidation, and pleural effusion classes. The whole-lung masks for this data set are provided separately. Thus, we combined them with the previous three classes to create labels of four classes.

In all previous data settings, a patient-wise random splitting strategy was considered to avoid any overlaps, such that all slices for a specific volume/patient appear in one splitting. All images are resized to the dimension of 256 × 256, where the resolution is 1.5 mm for the brain images, and in the range of ∼ 0.86 to 1.2 mm for the lung images. The intensity values normalized to [0, 1].

3.2 | Baselines

Our baselines include: (i) The lower bound models, which trained on the labeled volumes. (ii) The SSL models, which trained on the labeled and the unlabeled volumes. (iii) The upper bound models, which trained on the labeled volumes and the nine unlabeled volumes. However, all labels are revealed. (iv) Regularized ROAM: to evaluate our contributions, our method is introduced as a regularizer to the fully supervised lower and upper bound models, denoted as ROAM-LB, and ROAM-UB, respectively. For the SSL setting, the following methods are selected. (i) Bai et al. [30], (ii) Baur et al. [25], (iii) Cui et al. [36], and (iv) Zhang et al. [20]. We opt for these methods based on the following criteria. First, one method from each of the SSL approaches is chosen. Second, the easiness of implementation and the compatibility with the unified architecture. Third, we rule out the 3D methods or the methods that introduce sophisticated training mechanisms, such as multi-view training, uncertainty estimations, and domain adaptation.

3.3 | Implementation details

2D U-Net [58] is employed as backbone architecture, where the 2D slices are the input for the network. The weights are initialized using Xavier initialization and trained using Adam optimizer. The learning rate, weight decay, and batch size are set to 0.0001, 0.0001, and 8, respectively. The initial models denoted lower bounds trained for 40 epochs, the other semi-supervised, and the upper bound models further trained for an additional 40 epochs. The hyper-parameters are set to $T = 0.5, \alpha = \{0.75, 1\}$, and $\beta = \{75, 1\}$ for the brain and lung datasets, respectively. The mixup layer κ is selected randomly from the input, the first, and the last convolution layers, which is denoted as $\kappa = \{0, 1, L\}$ for the brain images and $\kappa = \{\Phi, 0, 1, L\}$ for the lung images, where Φ means no mixing of the data performed. All the experiments are performed using PyTorch framework hosted on an NVIDIA GTX 1080 8GB machine. The training time is about 6 h. The model with the best validation accuracy is used to report the testing results. Our code is available at ROAM.

3.4 | Evaluation metrics

The statistical summary of the Dice score [59], in addition to the Hausdorff distance (HD) [60], and the mean surface distance (MSD) [61], are reported. Dice score measures the overlapped area between the ground truth and the prediction divided by the overall area of prediction and the ground truth, Equation (10). The distance metrics measure the deviation between the outer surfaces S and S' of the segmentations Y and Y', such that the distance between a point s on surface S and the surface S' is given by the minimum of the Euclidean distance $d(s, S') = \min_{s' \in S'} \|s - s'\|_2$. Calculating this for all pixels gives the total distance between the surfaces S and S': $d(S, S')$. Now, the largest difference between the surface distances is defined as the Hausdorff distance (HD) and calculated as HD $= \max[d(S, S'), d(S', S)]$. The MSD, on the other hand, measures the average variation between the surfaces, that is, the segmentation and the GT, and is given as Equation (11). Note that we follow the One versus ALL methodology for calculating previous metrics such as for the multi-class segmentation, the mean value of any metric, that is, Dice, HD, or MSD is calculated by taking the value of each class individually and averaging them.

$$\mathrm{DSC}(p_i, y_i) = \frac{2\sum_j p_i g_i}{\sum_j p_i^2 + \sum_j g_i^2}. \qquad (10)$$

$$\mathrm{MSD} = \frac{1}{n_s + n_{s'}} \left(\sum_{s=1}^{n_s} d(s, S') + \sum_{s'=1}^{n_{s'}} d(s', S) \right), \qquad (11)$$

where n_s and $n_{s'}$ are the number of pixels for the surfaces S and S', respectively. A relative improvement (RI) *w.r.t* the baseline is also reported such that RI of a over b is : $(a - b)/b$.

3.5 | Ablation study

ROAM introduces the sharpening and concatenation operations to the Manifold Mixup. Also, it involves a set of hyper-parameters, that is (α, β) and design choices, that is $\varkappa$ in the training process. Thus, for the model selection, an ablation study and sensitivity analysis are conducted. In all these experiments, the training is done for 80 epochs, where the model with the highest validation accuracy is selected to report the testing results. The results are presented in Table 1.

3.5.1 | The selection of the random layer $\varkappa$

First, a set of layers are examined to realize on which the mixup operation will obtain the best results. That includes the input layer, the hidden layers, and a no-mix option, where the data passed to the network as per the usual training procedure. Note that $\varkappa$ is investigated when α and β are equal to 0.75 and 75, respectively. Please refer to Section 3.5.3 for why these values were selected. It is seen from the results in Table 1 that mixing the data at different random layers achieves better results than using only one fixed layer, except for $\varkappa = 2$. This correlation emphasizes the importance of alternating the hidden space with the input space during the training process, which provides the model with novel variations of the data that can not be generated using either the input or the hidden layers. Based on these results, $\varkappa = \{0, 1, L\}$ is fixed before the selection of the other parameters is investigated as presented in the next sections.

TABLE 1 Mean Dice for brain validation and testing datasets. ROAM, with $\varkappa = \{0, 1, L\}$, sharpening, concatenation, $\alpha = 0.75$, and $\beta = 75$, obtains the best validation results, hence, will be our model selection

Ablation	Value	Validation	Testing
ROAM	{0, 1, L}	**0.898**	0.870
$\varkappa$	0	0.881	0.852
	1	0.867	0.843
	2	0.894	0.872
	3	0.868	0.825
	4	0.863	0.828
	5	0.877	0.847
	L	0.865	0.843
	{0, 2, L}	0.884	0.851
	{1, 2, L}	0.883	0.863
	{0, 1, 5}	0.881	0.860
	{Φ, 0, 1, L}	0.882	0.864
	{All}	0.882	0.858
α	0.25	0.880	0.851
	2	0.885	0.836
β	0	0.893	0.844
Sharpening(✓)	Concatenation(✗)	0.878	0.850
Sharpening(✗)	Concatenation(✓)	0.861	0.823
Sharpening(✗)	Concatenation(✗)	0.870	0.843

Abbreviations: Φ, no data mixup; All, all hidden layers; L, last layer. First, $\varkappa$ is examined when α and β are equal to 0.75 and 75, respectively. Based on the results, $\varkappa = \{0, 1, L\}$ is used before the selection of the other parameters is investigated.

3.5.2 | The concatenation and the sharpening operations

In this experiment, we study removing the sharpening step on the soft labels and(or) concatenation of labeled and unlabeled batches, which resulted in three combinations as shown in the last rows in Table 1. Overall, a drop in the Dice score was observed when removing one or both steps. Yet, the worst result was obtained when applying the mixup operation on a concatenated batch without the sharpening. That is attributed to the mixing of the initial labels without minimizing their entropy through the sharpening step, which could harm the quality of the mixed-up data.

3.5.3 | The hyper-parameters α and β

First, three values of $\alpha = \{0.25, 0.75, 2\}$ are examined, where $\alpha = 0.75$ as in reference [49], $\alpha = 0.25$ to favor one sample over the other, and $\alpha = 2$ to make more balance between the different samples. It is noticed from Table 1 that ROAM obtains the best results when selecting $\alpha = 0.75$ because this value makes the mixed-up data closer to the original data while maintaining the novelty of the generated points. In the final part of our analysis, two values of $\beta = \{0, 75\}$ are investigated, where $\beta = 75$ as in reference [49], and $\beta = 0$ to evaluate the effectiveness of the

TABLE 2 Mean (Median) ± Std. of different evaluation metrics are reported on the MALC testing set for baselines and different SSL methods, including ours

Model Name	Dice coefficient ↑	RI (%) ↑	HD ↓	MSD ↓
Lower Bound	0.747(0.769) ± 0.071*	0	4.16 ± 0.43	1.06 ± 0.088
ROAM-LB	**0.823(0.841) ± 0.052**	**10.17**	**4.07 ± 0.35**	**1.05 ± 0.071**
Bai et al. [30]	0.800(0.815) ± 0.055*	7.10	4.06 ± 0.43	1.02 ± 0.086
Zhang et al. [20]	0.819(0.851) ± 0.060*	9.64	4.02 ± 0.44	1.00 ± 0.089
Cui et al. [36]	0.829(0.847) ± 0.045*	11.00	3.97 ± 0.38	1.03 ± 0.089
Baur et al. [25]	0.778(0.795) ± 0.071*	4.15	4.06 ± 0.40	1.05 ± 0.082
ROAM ($\varkappa = 0$)†	**0.852(0.866) ± 0.037**	**14.05**	**3.91 ± 0.35**	**0.99 ± 0.067**
ROAM ($\varkappa = 2$)	**0.872(0.881) ± 0.024**	**16.73**	**3.78 ± 0.28**	**1.00 ± 0.077**
ROAM ($\varkappa = \{0, 1, L\}$)	**0.870(0.873) ± 0.023**	**16.50**	**3.87 ± 0.31**	**1.00 ± 0.061**
Upper Bound	0.871(0.886) ± 0.044*	16.60	3.72 ± 0.42	0.95 ± 0.087
ROAM-UB	**0.893(0.902) ± 0.024**	**19.54**	**3.56 ± 0.34**	**0.91 ± 0.075**

Abbreviations: *, significant improvement; L, last layer; †, MixMatch [49]. ↑ (↓): The higher (lower) the better

newly-generated data on the training where we do not propagate the unlabeled loss. The results in Table 1 show that ROAM makes use of the unlabeled loss effectively, where the accuracy when $\beta = 75$ is better than the accuracy when $\beta = 0$. Furthermore, the obtained results at $\beta = 0$ show that the ROAM operation boosts the performance without the unlabeled loss. That is because the mixup between the labeled and the unlabeled examples augments the model with new virtual data.

In summary, the above analysis shows that ROAM, with $\varkappa = \{0, 1, L\}$, sharpening, concatenation, $\alpha = 0.75$, and $\beta = 75$, obtains the highest validation accuracy. Unless stated otherwise, we opt for these selections in the next experiments. In some experiments, we report the results at the input space, that is, ROAM($\varkappa = 0$) to compare our method with MixMatch. Also, we report the results for ROAM($\varkappa = 2$) because it obtains the second-highest validation accuracy, and to evaluate our method at the Manifold Mixup.

On the other hand, model selection experiments on lung validation data are conducted. Similarly, the hyper-parameters $\{\varkappa, \alpha, \beta\}$, concatenation, and sharpening steps are examined. The results in these experiments show that ROAM with $\varkappa = \{\Phi, 0, 1, L\}$, sharpening, concatenation, $\alpha = 1$, and $\beta = 1$, obtains the highest validation accuracy. Thus, we opt for this selection for lung segmentation testing results.

Taken together, the ablation study from both datasets shows the essential role of each component of our method on the segmentation task justifying its design choice. The testing results will be presented in the next section, while further analysis of the hyper-parameter tuning will be discussed in Section 6.

4 | WHOLE-BRAIN SEGMENTATION RESULTS

4.1 | Comparison with SSL methods

Table 2 illustrates the results for whole-brain segmentation. It is apparent from this table that our method outperforms the lower bound, upper bound, and all SSL methods with a statistical significance ($p < 0.001$).

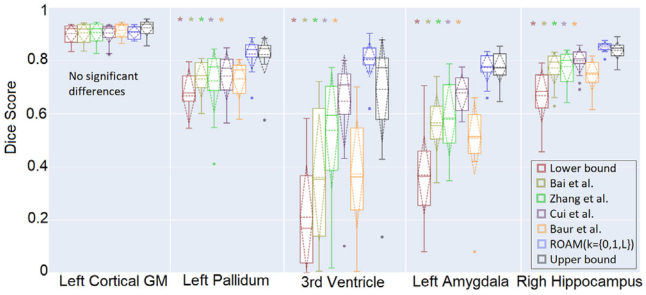


FIGURE 3 Dice score for selected structures. Our method significantly outperforms all other SSL methods in most brain structures

The best result, with average Dice of 87.0% and RI about 16.50%, is obtained by ROAM($\varkappa = \{0, 1, L\}$). Further analysis shows that ROAM($\varkappa = \{0, 1, L\}$) outperforms its variant ROAM($\varkappa = 0$), which is similar to MixMatch. The justification is that ROAM($\varkappa = \{0, 1, L\}$) introduces a lot of variations and generates novel data points that have never been seen before via its ROAM. Thus, it avoids overfitting.

Interestingly, a similar performance is reported for ROAM($\varkappa = 2$).

Further statistical tests revealed that our method achieves the best HD and MSD scores of 3.87 and 1.00, respectively. Moreover, ROAM-LB and ROAM-UB models outperform their competitors significantly with average Dices of 82.3% and 89.3%, and RI of 10.17% and 19.54%, respectively. That is strong evidence that applying ROAM as a regularizer provides the model with new data points. Consequently, it boosts the performance without the need for any additional data. Surprisingly, ROAM-LB outperforms most SSL methods by significant margins, confirming its advantages as a strong regularizer.

4.2 | Structure level results

The segmentation results for some internal structures are reported in Figure 3. The results show that ROAM significantly outperforms all SSL methods in most structures. Besides,

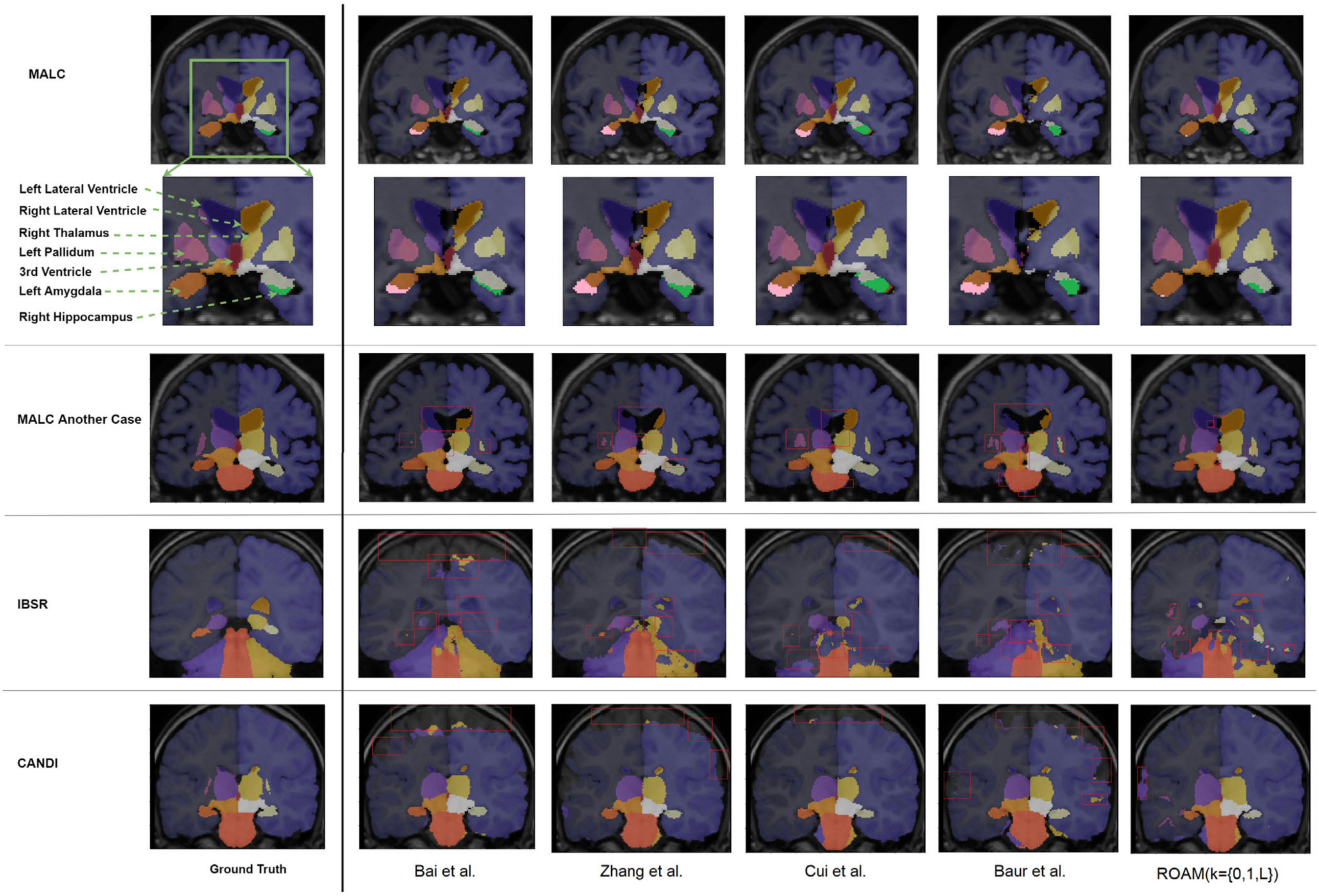


FIGURE 4 Qualitative results of brain segmentation. The first row shows a coronal view of one case from the MALC dataset. The second row shows a cropped version highlighting selected structures for the same case. Another case is shown in the third row. Further, the segmentation results in the presence of domain shift are shown in the fourth and fifth rows of IBSR and CANDI datasets, respectively. In these cases, ROAM obtains the best results, where the red boxes show the false predictions made by different models

ROAM excels over the upper bound in the Right Hippocampus and 3rd Ventricle.

Additionally, the performance of our method is consistent across different structures. That is clearly shown in the Left Pallidum, 3rd Ventricle, Left Amygdala, and Right Hippocampus. Our model achieves a lower performance in Left Cortical GM, yet the difference is not statistically significant.

4.3 | Qualitative results

To provide more insights on the performance, the qualitative results are shown in Figure 4. The first row shows the predictions on the MALC dataset. The second row shows a cropped version, where we highlighted the right and left lateral ventricle, right thalamus, right hippocampus, left palladium, left amygdala, and 3rd ventricle. Despite the complexity of these small structures, ROAM performs more reliably than all SSL methods. To support our findings, we also include another case from the MALC dataset in the third row. Likewise, ROAM surpasses all SSL methods. Finally, the predictions under cross-domain settings are shown for IBSR and CANDI datasets in the fourth and fifth rows, respectively. In general, ROAM predicts more accurate results than other SSL methods indicating its generalization ability to other domains. Together, the quantitative and the qualitative results show the superiority of ROAM against all SSL methods.

4.4 | Comparison with SOTA for whole-brain segmentation

To realize the effectiveness of ROAM in a fully supervised fashion, we run our method using the labeled data. In this experiment, the batch is mixed with its permuted version, where no sharpening nor pseudo labeling steps are performed. Also, β is set to 0 so that the unsupervised loss is not propagated. The MACL dataset is used for the training for 80 epochs, where the model at the last epoch is saved. Our method is compared with U-Net[58], and QuickNAT [15]. In contrast to U-Net and our model, QuickNAT is pre-trained using 581 labeled volumes from IXI dataset. Table 3 shows the testing

TABLE 3 Dice score for fully supervised models. ROAM significantly outperforms both U-Net and on par with QuickNAT without sophisticated pre-training mechanism

Model name	Mean (median) ± std	RI (%)
U-Net	0.874(0.888) ± 0.039	0
QuickNAT	0.895(N/A) ± 0.055	2.40
ROAM ($\varkappa = 0$)	0.890(0.898) ± 0.025	1.83
ROAM ($\varkappa = \{0, 1, L\}$)	**0.895(0.901)** ± **0.022**	2.40
ROAM ($\varkappa = 2$)	**0.897(0.906)** ± **0.025**	2.63

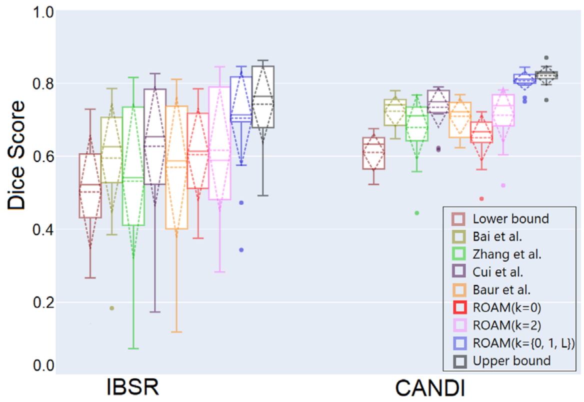


FIGURE 5 Domain shift results. The domain shift has a lower effect on ROAM than the other methods

results on the MALC dataset. All ROAM variations significantly outperform U-Net and are on par and sometimes outperform QuickNAT, without a sophisticated pre-training mechanism. Note that ROAM ($\varkappa = 0$) is a special case of our method where the mixup is performed at the input space, that is, *MixMatch*. Further, the results show that our models achieve lower standard deviations compared to other methods. In summary, the results show that our simple but elegant ROAM operation leads to SOTA results without the need for large datasets.

4.5 | Realistic evaluation of ROAM

The purpose of the next set of experiments is to (i) assess ROAM in the presence of domain shift, (ii) show the correlation between the amount of labeled and unlabeled data on the overall performance, following the recommendations of reference [50].

4.5.1 | Domain shift results

The trained models were picked and tested on IBSR and CANDI datasets. The results in Figure 5 show a drastic drop in all models, including the baseline ones. This drop is higher on the ISBR dataset. However, ROAM($\varkappa = \{0, 1, L\}$) performs just as well in both cases and is less sensitive to the domain shift problem compared with other models, including ROAM($\varkappa = 2$) and ROAM($\varkappa = 0$). Surprisingly, although ROAM($\varkappa = 2$) achieves one of the best results on the MALC dataset, it has less generalization ability than ROAM($\varkappa = \{0, 1, L\}$). The results indicate that the domain shift has a lower effect on ROAM than the other methods.

4.5.2 | Changing amount of labeled data

At first, we fix the number of unlabeled data at 1500 slices while gradually increasing the amount of labeled data from 100 to 500. With successive increases in the amount of the labeled data, our model displayed a higher performance and confidence compared to other models (cf. Figure 6a). This confidence level is inconsistent in other models.

The same superiority is also observed at the lowest amount of labeled data (100 slices), where the obtained Dice scores are 0.622, 0.402, 0.500, 0.571, 0.400 for ROAM, Bai et al. [30], Zhang et al. [20], Cui et al. [36], and Baur et al. [25], respectively, cf. Figure 6a, the results on the far left.

4.5.3 | Changing amount of unlabeled data

In this experiment, we fix the labeled data at 500 slices while gradually reducing the unlabeled from 1500 to 500. The results are shown in Figure 6b. In contrast to other methods, our model shows its superior *w.r.t* variable amount of unlabeled data.

The figure shows that our approach still outperformed when the amount of unlabeled data is the lowest (500 slices) with considerable margins. The obtained Dice scores for ROAM against the other methods are 0.820, 0.795, 0.798, 0.809, and 0.760, respectively, cf. Figure 6b, the results on the far right.

Yet, [36] achieves insignificant higher Dice at 1000 unlabeled slices.

Both results confirm the superiority of our method at a low data regime.

5 | LUNG SEGMENTATION RESULTS

In the second part of our experiments, ROAM is validated on lung CT images for lung segmentation. Note that our model selection for this dataset is ROAM($\varkappa = \{\Phi, 0, 1, L\}$), α and β = 1.

5.1 | COVID-19-CT-seg-Benchmark results

5.1.1 | Quantitative results

The segmentation results, reported in Table 4, show that ROAM and ROAM-LB surpass their competitors in the overall results, see the foreground column in Table 4. The obtained relative improvements are 10.86%, 17.09%, and 18.09%, respectively, for

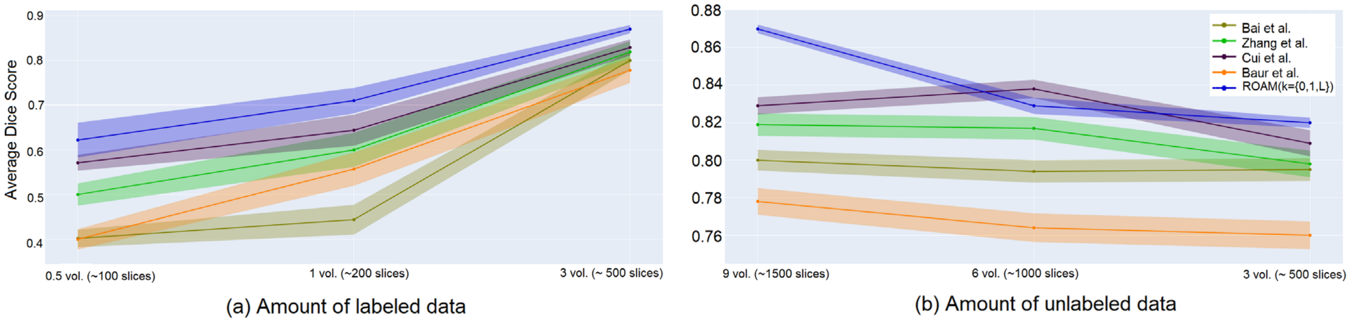


FIGURE 6 Varying amount of data. The shaded region represent the standard deviation. The more labeled or unlabeled data being used, the higher the performance and confidence of our model compared to others

TABLE 4 Lung CT images segmentation results. ROAM outperforms SSL methods for the infection and lung classes, while it outperforms the lower bound in the overall and lung results. ROAM shows lower performance in the infection segmentation comparing to U-net. The foreground column includes the infection, the left, and the right lung classes. (), negative value

		Foreground		Infection		Lung	
Setting	**Model**	**Mean (median) ± std**	**RI(%)**	**Mean (median) ± std**	**RI(%)**	**Mean (median) ± std**	**RI(%)**
Lower bounds	U-Net	0.702(0.738) ± 0.176	0	**0.543(0.657) ± 0.254**	0	0.782(0.897) ± 0.231	0
	ROAM-LB	**0.777(0.839) ± 0.126**	10.68	0.528(0.606) ± 0.275	(2.76)	**0.902(0.942) ± 0.121**	15.35
SSLs	Bai et al. [30]	0.730(0.772) ± 0.154	3.99	0.552(0.599) ± 0.233	1.66	0.819(0.881) ± 0.153	4.73
	Zhang et al. [20]	0.736(0.775) ± 0.161	4.84	0.606(0.717) ± 0.251	11.60	0.802(0.880) ± 0.213	2.56
	Cui et al. [36]	0.810(0.873) ± 0.116	15.38	0.605(0.672) ± 0.239	11.42	0.913(0.953) ± 0.102	16.75
	ROAM	**0.822(0.887) ± 0.122**	17.09	**0.632(0.710) ± 0.252**	16.39	**0.918(0.957) ± 0.103**	17.39
Upper bounds	U-Net	**0.849(0.888) ± 0.096**	20.94	**0.675(0.737) ± 0.229**	24.31	**0.936(0.974) ± 0.091**	19.69
	ROAM-UB	0.829(0.872) ± 0.107	18.09	0.630(0.686) ± 0.218	16.02	0.929(0.974) ± 0.102	18.80

ROAM-LB, ROAM, and ROAM-UB. In line with the whole-brain segmentation results, it also observed that ROAM-LB outperforms the other SSL methods by considerable margins. In contrast to that, ROAM-UB performs just lower than the upper bound. Surprisingly, ROAM-LB's segmentation score for the infection dropped by 2.76% compared to the U-Net. In summary, ROAM outperforms all SSL methods for all classes, and outperforms the lower bound in the overall and lung results. Yet, ROAM shows lower performance in the infection segmentation when compared to U-net. Further discussion is presented in Section 6.1.

5.1.2 | Qualitative results

The segmentation predictions for the previous models are shown in Figure 7. The first two columns in the first row show the input image with its ground truth. The next four columns present the segmentation results for the lower and upper bounds, respectively. The second row shows the predictions for the SSL methods and ROAM. The red boxes are drawn to show the false predictions made by different models. Except for the upper bound, ROAM makes fewer false positives and generates more accurate predictions than the other models in all settings. Moreover, ROAM-LB performs better than U-Net lower bound and better than some SSL methods such as in references [30] and [20].

5.2 | MedSeg: Cross domain and class mismatch results

MedSeg dataset consists of 100 CT images divided into 80 training images and 20 validation images, with four classes of lung, ground-glass opacity, consolidation, and pleural effusion. In this experiment, the model trained on MedSeg while it was tested on the COVID-19-CT-Seg-Benchmark dataset. Notice that the last dataset contains segmentation of the right lung, left lung, and infection classes. Thus, the goal of this experiment is to investigate the ability of ROAM cross domains and class mismatch conditions. Note that the training and the testing images come from different datasets with a domain shift problem. Further, the training and testing classes differ, making it a very challenging task.

To resolve this issue, we perform two steps. First, after training the models, we generate the four-class predictions. Then, we assemble the predictions of ground-glass opacity, consolidation, and pleural effusion as one class called the infection class, yet the lung predictions remain without any modification. The result from the previous step is predictions of two classes; lung

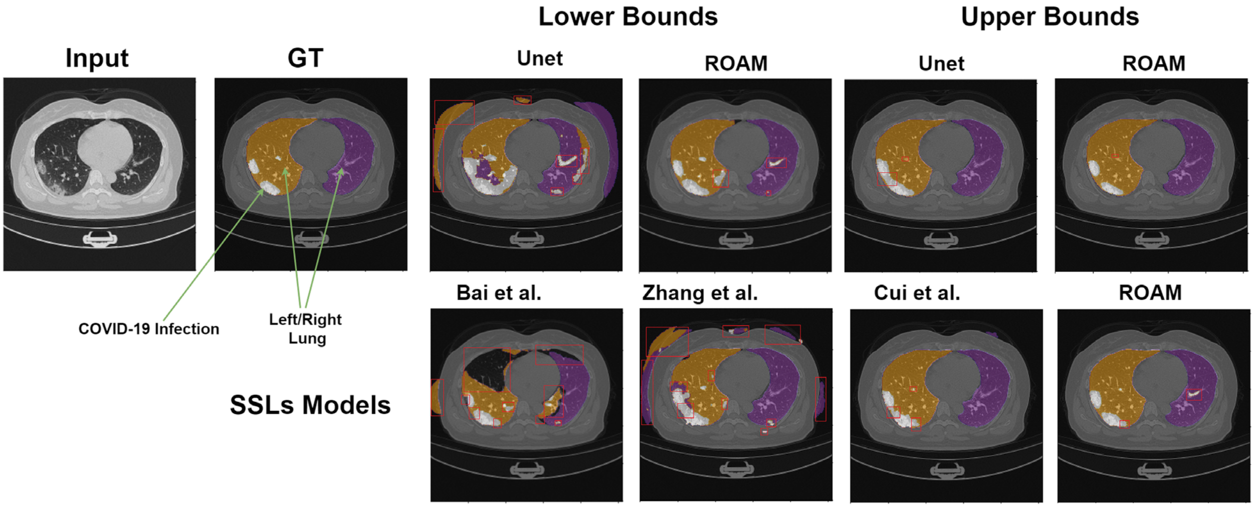


FIGURE 7 Qualitative results of lung segmentation. Red boxes represent the false positives. ROAM generates more accurate predictions than the other models with the exception of the U-Net-UB

TABLE 5 Cross domain and class mismatch results. The models are trained on MedSeg dataset, while tested on COVID-19-CT-Seg-Benchmark dataset. ROAM enhances the prediction of the baseline

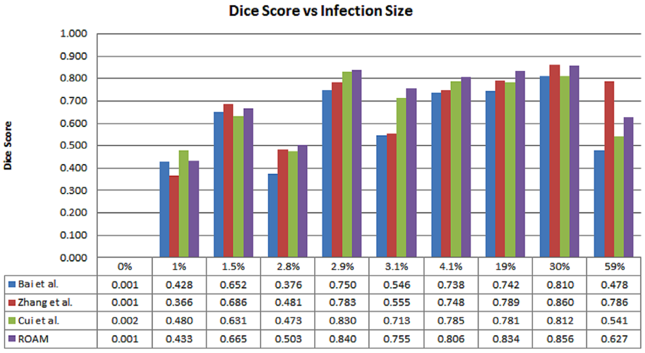
	Foreground		Infection	
Model	Mean (median) ± std	RI(%)	Mean (median) ± std	RI(%)
U-Net	0.675(0.684) ± 0.107	0	0.449(0.496) ± 0.233	0
ROAM	**0.714(0.728) ± 0.111**	5.78	**0.522(0.501) ± 0.224**	16.26

and infection. The second step, however, is performed on the testing data. Specifically, the right and lung masks are assembled as one class called the lung class, while the infection masks remain without any modification. The result from this step is labels with two classes; lung and infection. Having performed these two steps, our results for this experiment are generated.

The results are reported in Table 5. We notice that ROAM enhances the predictions by 16.26% and 5.78% for the infection and foreground, respectively. The results of this experiment are in line with the results reported for brain images. That is, both are consistent and highlight the ability of ROAM to generalize to unseen data.

5.3 | Performance versus infection sizes

Thus far, the performance of ROAM at different data settings, domain shift, and class mismatch has been reported. In this experiment, we try to analyze the effect of the COVID-19 infection size *w.r.t* the lung size. Figure 8 shows the individual Dice score for each test volume from the COVID-19-CT-Seg-Benchmark dataset. The percentage below each column represents the infection size. It stands out that the same pattern for all SSLs models is found. First, when the infection size is below 3%, all SSL methods produce uncertain results. Second, the best results are obtained when the classes are balanced (at 30%). In this case, the infection represents one third of the lung size, while the remaining percentages are one third for the left lung and one third for the right lung. Third, the results at 59% represent outlier cases that fool all models because the infection represents the minor class in the image in the usual cases. In general, ROAM obtains the best results regardless of the infection percentage.



	0%	1%	1.5%	2.8%	2.9%	3.1%	4.1%	19%	30%	59%
🟦 Bai et al.	0.001	0.428	0.652	0.376	0.750	0.546	0.738	0.742	0.810	0.478
🟥 Zhang et al.	0.001	0.366	0.686	0.481	0.783	0.555	0.748	0.789	0.860	0.786
🟩 Cui et al.	0.002	0.480	0.631	0.473	0.830	0.713	0.785	0.781	0.812	0.541
🟪 ROAM	0.001	0.433	0.665	0.503	0.840	0.755	0.806	0.834	0.856	0.627

FIGURE 8 Dice versus infection. The *x*-axis represents the percentage of the infection size to the lung size. When the percentage below 3%, SSLs produce uncertain results. The best obtained when the classes are balanced (at 30%). The percentage of (59%) is an outlier case

6 | DISCUSSION

This paper proposes ROAM as an SSL method that utilizes the modern regularization methods, that is, MixUp and Manifold Mixup to boost the model with newly generated data points. Our method overcomes the limitations of the previous works by exploring the manifold and performing the linear interpolation at a randomly selected subset of input and hidden layers. Consequently, it generates new data points and additional training signals that suit the complexity of medical image segmentation for SSL. Moreover, our method utilizes the better organized hidden representation of classes and produces consistent soft labels for the corresponding data points generated via the mixup operation.

6.1 | ROAM performance across different datasets

Our method is validated using five publicly available datasets for the brain and lung images. These datasets are heterogeneous. While the structures in the brain images are almost rigid and geometrically constrained, the lung images contain highly variable sizes of COVID-19 infection. The results show that ROAM performs consistently and outperforms all SSL methods with large margins across these datasets. Further, the robustness of ROAM is significant in the brain segmentation, where ROAM always obtains the best results. The main advantage of ROAM is the generating of new virtual data points. This process enriches our method with a wide range but free training signals which are explored not only in the input space but also at the hidden representations. Further, what makes our new data beneficial is selecting the mixing factor λ, which has been selected to keep the virtual points in the vicinity distribution of the training examples. In contrast, the other methods are limited to the original training examples such that whatever approach is used, the knowledge gain is still limited. Another advantage is that our method enhances the quality of the pseudo labels by the sharpening operation, while none of the remaining approaches make use of this post-process step. However, one limitation has been noticed in the lower bound for the COVID-19 infection segmentation. Even though ROAM enhances the overall prediction, it fails to enhance the segmentation of the infection class. This could be attributed to the fact that mixing highly imbalanced classes, that is, infection pixels versus lung pixels, at a low data regime, could bias the model to the dominant class. In another anticipated finding, ROAM-UB achieves lower performance than the upper bound baseline model. A possible explanation might be that the amount of data—at the upper bound setting—is enough for the training. Therefore, augmenting the model with additional virtual data points may not be useful. Also, it has been shown that our lower bound model (ROAM-LB) consistently outperforms many SSL methods. That means this regularization technique, with just a few labeled data, could surpass the other SSL methods which have access to a large amount of unlabeled data. Moreover, ROAM generates new data points through its linear interpolation. The effectiveness of this operation is essential at a lower data regime where the data is crucial for the training. Interpretability might be another limitation of our method. The generated data from the mixup operation could be hard to interpret, especially when the two mixed-up samples are randomly selected. Consider, for example, in brain experiments, an image containing the white matter was mixed with another one containing grey matter or any other brain structure. For a human or an expert, the resulting image will not be recognized as a known structure in the brain. Thus, instead of augmenting the training, this should confuse the model. Although, our experiments showed that this operation boosted the performance, the explainability of our method needs further investigation.

6.2 | ROAM($\kappa = 2$) results

Interestingly, ROAM ($\kappa = 2$) achieves one of the best results on the MALC dataset. We attributed this to hidden representation at this layer, where it might be the most organized and concentrated among all layers. Thus, the inconsistency in soft labels is minimized. Despite that, ROAM ($\kappa = 2$) has less generalization ability than ROAM ($\kappa = 0, 1, L$), indicating the possibility of overfitting to the training data. Further investigation might lead to more explanations.

6.3 | Generalizability and domain mismatch

One way to alleviate the need for a large amount of annotated data is to utilize datasets generated from different sources. Usually, these datasets come with many challenges, that is, different cohorts, scanning protocols, and scanners. That leads to a technical challenge, the so-called domain shift. This problem has been investigated in this paper and have noticed that all SSL methods, including ROAM, suffer in the presence of domain shift. Yet, ROAM was less sensitive, see Figure 5 and Table 5. Nevertheless, we make no claim here that our approach is domain agnostic. Thus, further research in handling the domain shift in the SSL methods is of high importance.

6.4 | Convergence

Manifold Mixup is guaranteed to be converged when the mixup operation is performed at a hidden layer, as long as the dimensionality of that layer is greater than the number of the classes [44]. Here, this condition is satisfied where the dimensionality of the hidden layers > 32, which is greater than the number of segmentation classes, that is, 28 for brain images and 4 for lung images.

6.5 | Handling skip connections

An important question is how to handle the skip connections when mixing at a random layer. Do the skip connections get interpolated using the same lambda as the convolution layers or just forwarded without any mixup? For example, when mixing two samples x_1 and x_2 at a random hidden layer, that is, $\kappa = 2$, the skip connections related to that layer still hold the original data from the first hidden layer. Therefore, they will not correspond to the mixed-up labels properly, which might cause a problem. One suggestion to handle this issue is to perform the mixup for a given layer and the skip connections up to that layer with the same lambda and the same example pairing. Practically, we investigate this solution on MALC and COVID-19 datasets when $\kappa = 2$, and report the results in

Table 6. It is shown that ROAM performs differently, and no such approach produces consistent performance. For instance, the skip-connections mixup at SSL settings impairs the results while it has almost no effect or a negligible positive effect at the upper bounds. The issue of handling the skip connections is intriguing and could be usefully explored in further research. Fortunately, this problem did not happen in our main scenarios, that is, performing random mixup at $\kappa = \{0, 1, L\}$ because the mixed labels correspond to the mixed data as well. Yet, it is not the case for the manifold mixup when $\kappa = 2$, which surprisingly shows a superior result. One of the reasons could be attributed to the choice of the beta distribution parameter, that is, α. For instance, when α is less than 1, the mixed data tend to preserve the original data point. Therefore, performing manifold at the bottleneck or other layers might not have such an expected negative impact.

6.6 | Infection size

ROAM can be affected by the highly imbalanced dataset as can many SSL methods. Figure 8 shows that the best performance is obtained when the classes are equally distributed. Thus, performing mixup operations with highly imbalanced data remains a challenging question. Our preliminary analysis in this direction paves the way for further investigation.

6.7 | Validation datasets

One problem of using small validation datasets is the inconsistency of the results, which may not reflect the actual performance of the model [50]. The smaller the validation set, the larger the variations in the output. Moreover, reference [50] also argued that a comparison between different SSL models is possible when the validation set is equal to the training one. Here, we consider all these recommendations in our implementation. Consequently, the reported results fairly reflect the actual performance of each model.

TABLE 6 The results for whole-brain and lung segmentation at $\kappa = 2$ with/out skip-connections mixup. ROAM works better without skip-connections mixup at SSL setting, while it performs just lower at the upper bounds

Dataset	Model	SK	Mean (median) ± std
Brain	ROAM-SSL	✓	0.834(0.853) ± 0.047
		×	0.872(0.881) ± 0.024
	ROAM-UB	✓	0.892(0.898) ± 0.024
		×	0.890(0.898) ± 0.023
Lung	ROAM-SSL	✓	0.779(0.849) ± 0.137
		×	0.797(0.860) ± 0.121
	ROAM-UB	✓	0.851(0.889) ± 0.100
		×	0.850(0.901) ± 0.107

Abbreviation: SK, skip-connection miuxp

6.8 | The unsupervised loss

Interestingly, $\beta = 0$ shows the third-highest validation results. β is a hyperparameter that controls the contribution of the unsupervised loss. Setting $\beta = 0$ implicitly means that our model is still augmented with new data points from our random mixup yet without the unlabeled single. Based on the selected λ in Equations (7) and (8), the newly generated data points are close to the labeled data. In other words, the new data are in the vicinity distribution of the labeled data, that is, high-quality data is generated, justifying the boost in the performance. On the other hand, setting $\beta > 0$ means that we propagate the training signals from the unlabeled data. These signals might be noisy and introduce uncertainty to the model because of the low quality of the pseudo labels. Hence, a decrease in performance was observed. However, after applying the sharpening, an enhancement is noticed in the model performance because the sharpening operation helps to generate more accurate pseudo labels, as shown in Figure 1d.

6.9 | Hyper-parameters tuning

ROAM involves a set of hyper-parameters and design choices besides the standard ones. Although fine-tuning such an amount of parameters is a tedious task, our results show that ROAM outperforms all SSL methods in a wide range of hyper-parameter choices. Thus, with a little effort, one can achieve SOTA performance. Our argument can be supported by the following examples. First, ROAM outperforms all SSL models regardless of the selected layer κ. Also, the lowest scores obtained by ROAM, when $\kappa = \{3 \text{ or } 4\}$, are better than all other SSL methods, with one exception of reference [36]. Third, all ROAM variations, that is, the sharpening and concatenation steps, outperform all other SSL models. Fourth, we show that the newly generated data boosts the performance without the need for the unlabeled loss when $\beta = 0$.

That is, the number of hyper-parameters can be reduced significantly by fixing $\kappa = \{0, 1, L\}$ and just fine-tuning α and β, which is the standard procedure in many SSL methods. Consequently, our method does not require any extra effort or exhausting design choices.

Based on that, our approach is easy to implement and can be generalized to different datasets, which have been shown in the brain and lung segmentation. Further, our code is publicly available for benchmarking and reproducibility.

7 | CONCLUSION

Here, we propose ROAM for SSL in medical images. ROAM takes the advantages of both MixMatch and Manifold Mixup to boost the performance of the model with new generated data points that fit the complexity of the medical images. While both methods depend on either the input layer or the hidden representations to generate new data points, our method makes use of a random combination of these layers. Our experiments show that ROAM is less prone to overfitting and has a better generalization property.

Our method shows a superior and SOTA performance on the whole brain, lung, and COVID-19 infection segmentation compared to other SSL methods. We tested ROAM in both supervised and semi-supervised settings and we have shown its preference against other approaches. Our comprehensive analysis shows that our method utilizes both labeled and unlabeled data efficiently, proving its stability, superiority, and consistency. Further, the mixup operation has been investigated at skip connections, in the U-net architecture, which has not been studied by any of the previous methods.

So far, the quality of the pseudo labels mainly depends on the initial guess and the mixup coefficient λ. However, one could think of modeling this coefficient as a function of uncertainty measures. Also, to generate more realistic mixed-up data, one could investigate performing the mixup operation on disentangled representations [62]. Our experiment demonstrates a robust performance of our method under domain shift. Nevertheless, domain invariant SSL methods should be further investigated. ROAM, as with other SSL methods, can be affected by the class-imbalance datasets. Instead of naive mixup, one could investigate more intelligent ways of data mixing.

ACKNOWLEDGEMENTS

Tariq Bdair is financially supported by the German Academic Exchange Service (DAAD). Shahid Albarquoni was supported by the PRIME programme of the German Academic Exchange Service (DAAD) with funds from the German Federal Ministry of Education and Research (BMBF). The authors thank Alex Lamb for his feedback on employing ROAM with skip connection.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data openly available in a public repository that issues datasets with DOIs.

ORCID

Tariq Bdair https://orcid.org/0000-0003-2049-2113
Benedikt Wiestler https://orcid.org/0000-0002-2963-7772
Nassir Navab https://orcid.org/0000-0002-6032-5611
Shadi Albarqouni https://orcid.org/0000-0003-2157-2211

REFERENCES

1. Suetens, P.: Fundamentals of Medical Imaging. Cambridge university press, Cambridge (2017)
2. Aerts, H.J., Velazquez, E.R., Leijenaar, R.T., Parmar, C., Grossmann, P., Carvalho, S., et al.: Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. Nat. Commun. 5(1), 1–9 (2014)
3. Sharma, N., Aggarwal, L.M.: Automated medical image segmentation techniques. J. Med. Phys. 35(1), 3 (2010)
4. Pham, D.L., Xu, C., Prince, J.L.: Current methods in medical image segmentation. Annu. Rev. Biomed. Eng. 2(1), 315–337 (2000)
5. Nikolov, S., Blackwell, S., Zverovitch, A., Mendes, R., Livne, M., De Fauw, J., et al.: Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy. arXiv:180904430 (2018)
6. Kickingereder, P., Isensee, F., Tursunova, I., Petersen, J., Neuberger, U., Bonekamp, D., et al.: Automated quantitative tumour response assessment of MRI in neuro-oncology with artificial neural networks: a multicentre, retrospective study. Lancet Oncol. 20(5), 728–740 (2019)
7. Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., et al.: U-net: deep learning for cell counting, detection, and morphometry. Nat. Methods 16(1), 67–70 (2019)
8. De.Fauw, J., Ledsam, J.R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., et al.: Clinically applicable deep learning for diagnosis and referral in retinal disease. Nat. Med. 24(9), 1342–1350 (2018)
9. Hollon, T.C., Pandian, B., Adapa, A.R., Urias, E., Save, A.V., Khalsa, S.S.S., et al.: Near real-time intraoperative brain tumor diagnosis using stimulated raman histology and deep neural networks. Nat. Med. 26(1), 52–58 (2020)
10. Chiu, S.J., Li, X.T., Nicholas, P., Toth, C.A., Izatt, J.A., Farsiu, S.: Automatic segmentation of seven retinal layers in SDOCT images congruent with expert manual segmentation. Opt. Express 18(18), 19413–19428 (2010)
11. Hu, S., Hoffman, E.A., Reinhardt, J.M.: Automatic lung segmentation for accurate quantitation of volumetric x-ray CT images. IEEE Trans. Med. Imaging 20(6), 490–498 (2001)
12. Maier.Hein, L., Eisenmann, M., Reinke, A., Onogur, S., Stankovic, M., Scholz, P., et al.: Why rankings of biomedical image analysis competitions should be interpreted with care. Nat. Commun. 9(1), 1–13 (2018)
13. Coupé, P., Mansencal, B., Clément, M., Giraud, R., deSenneville, B.D., Ta, V.T., et al.: Assemblynet: a novel deep decision-making process for whole brain MRI segmentation. arXiv:190601862 (2019)
14. Gu, Z., Cheng, J., Fu, H., Zhou, K., Hao, H., Zhao, Y., et al.: Ce-net: context encoder network for 2D medical image segmentation. IEEE Trans. Med. Imaging 38(10), 2281–2292 (2019)
15. Roy, A.G., Conjeti, S., Navab, N., Wachinger, C., Initiative, A.D.N., et al.: Quicknat: a fully convolutional network for quick and accurate segmentation of neuroanatomy. NeuroImage 186, 713–727 (2019)
16. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nat. Methods 18(2), 203–211 (2021)
17. Schmarje, L., Santarossa, M., Schröder, S.M., Koch, R.: A survey on semi-, self-and unsupervised learning for image classification. IEEE Access 9, 82146–82168 (2021)
18. Van.Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Mach. Learn. 109(2), 373–440 (2020)
19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
20. Zhang, Y., Yang, L., Chen, J., Fredericksen, M., Hughes, D.P., Chen, D.Z.: Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 408–416, Springer (2017)
21. Nie, D., Gao, Y., Wang, L., Shen, D.: Asdnet: Attention based semi-supervised deep networks for medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 370–378, Springer (2018)
22. Chen, S., Bortsova, G., Juarez, A.G.U., vanTulder, G., deBruijne, M.: Multi-task attention-based semi-supervised learning for medical image segmentation. arXiv:190712303 (2019)
23. Alizadehsani, R., Sharifrazi, D., Izadi, N.H., Joloudari, J.H., Shoeibi, A., Gorriz, J.M., et al.: Uncertainty-aware semi-supervised method using large unlabeled and limited labeled covid-19 data. ACM Trans. Multimedia Comput. Commun. Appl. 17(3s), 1–24 (2021)
24. Kamran, S.A., Hossain, K.F., Tavakkoli, A., Zuckerbrod, S.L., Baker, S.A.: VTGAN: Semi-supervised retinal image synthesis and disease prediction using vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3235–3245 (2021)
25. Baur, C., Albarqouni, S., Navab, N.: Semi-supervised deep learning for fully convolutional networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 311–319, Springer, Heidelberg (2017)
26. Ganaye, P.A., Sdika, M., Benoit Cattin, H.: Semi-supervised learning for segmentation under semantic constraint. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 595–602. Springer, Heidelberg (2018)

27. Ghorbani, M., Bahrami, M., Kazi, A., Soleymani-Baghshah, M., Rabiee, H.R., Navab, N.: GKD: Semi-supervised graph knowledge distillation for graph-independent inference. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 709–718. Springer, Heidelberg (2021)
28. Ghorbani, M., Kazi, A., Baghshah, M.S., Rabiee, H.R., Navab, N.: Ra-gcn: graph convolutional network for disease prediction problems with imbalanced data. Med. Image Anal. 75, 102272 (2022)
29. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in Neural Information Processing Systems, pp. 529–536 (2005)
30. Bai, W., Oktay, O., Sinclair, M., Suzuki, H., Rajchl, M., Tarroni, G., et al.: Semi-supervised learning for network-based cardiac MR image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 253–260. Springer, Heidelberg (2017)
31. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100 (1998)
32. Xia, Y., Yang, D., Yu, Z., Liu, F., Cai, J., Yu, L., et al.: Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation. Med. Image Anal. 65, 101766 (2020)
33. Bai, T., Zhang, Z., Zhao, C., Luo, X.: A novel pseudo-labeling approach for cell detection based on adaptive threshold. In: International Symposium on Bioinformatics Research and Applications, pp. 254–265. Springer, Heidelberg (2021)
34. Li, Y., Chen, J., Xie, X., Ma, K., Zheng, Y.: Self-loop uncertainty: a novel pseudo-label for semi-supervised medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 614–623. Springer, Heidelberg (2020)
35. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems, pp. 1195–1204 (2017)
36. Cui, W., Liu, Y., Li, Y., Guo, M., Li, Y., Li, X., et al.: Semi-supervised brain lesion segmentation with an adapted mean teacher model. In: International Conference on Information Processing in Medical Imaging, pp. 554–565. Springer, Heidelberg (2019)
37. Bortsova, G., Dubost, F., Hogeweg, L., Katramados, I., deBruijne, M.: Semi-supervised medical image segmentation via learning consistency under transformations. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 810–818. Springer, Heidelberg (2019)
38. 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: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 605–613. Springer, Heidelberg (2019)
39. Li, X., Yu, L., Chen, H., Fu, C.W., Xing, L., Heng, P.A.: Transformation-consistent self-ensembling model for semisupervised medical image segmentation. IEEE Trans. Neural Netw. Learn. Syst. 32(2), 523–534 (2020)
40. Li, Y., Luo, L., Lin, H., Chen, H., Heng, P.A.: Dual-consistency semi-supervised learning with uncertainty quantification for covid-19 lesion segmentation from ct images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 199–209. Springer, Heidelberg (2021)
41. Meyer, A., Ghosh, S., Schindele, D., Schostak, M., Stober, S., Hansen, C., et al.: Uncertainty-aware temporal self-learning (UATS): semi-supervised learning for segmentation of prostate zones and beyond. Artif. Intell. Med. 116, 102073 (2021)
42. Wang, X., Chen, H., Xiang, H., Lin, H., Lin, X., Heng, P.A.: Deep virtual adversarial self-training with consistency regularization for semi-supervised medical image classification. Med. Image Anal. 70, 102010 (2021)
43. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez Paz, D.: Mixup: beyond empirical risk minimization. arXiv:171009412 (2017)
44. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., et al.: Manifold mixup: better representations by interpolating hidden states. In: International Conference on Machine Learning, pp. 6438–6447 (2019)
45. Chaitanya, K., Karani, N., Baumgartner, C.F., Becker, A., Donati, O., Konukoglu, E.: Semi-supervised and task-driven data augmentation. In: International Conference on Information Processing in Medical Imaging, pp. 29–41. Springer, Heidelberg (2019)
46. Eaton-Rosen, Z., Bragman, F., Ourselin, S., Cardoso, M.J.: Improving data augmentation for medical image segmentation. In: International Conference on Medical Imaging with Deep Learning (2018)
47. Panfilov, E., Tiulpin, A., Klein, S., Nieminen, M.T., Saarakkala, S.: Improving robustness of deep learning based knee MRI segmentation: Mixup and adversarial domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
48. Jung, W., Park, S., Jung, K.H., Hwang, S.I.: Prostate cancer segmentation using manifold mixup U-net. In: International Conference on Medical Imaging with Deep Learning–Extended Abstract Track (2019)
49. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. Advances in Neural Information Processing Systems. 32 (2019)
50. Oliver, A., Odena, A., Raffel, C.A., Cubuk, E.D., Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. In: Advances in Neural Information Processing Systems, pp. 3235–3246 (2018)
51. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning. In: Chapelle, O., et al., (eds.) 2006. IEEE Trans. Neural Networks 20(3), 542 (2009)
52. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Vol 70. pp. 1321–1330 (2017)
53. Landman, B.A., Warfield, S.: MICCAI 2012: grand challenge and workshop on multi-atlas labeling. In: Proc. International Conference on Medical Image Computing and Computer Assisted Intervention, MICCAI. 2012 (2012)
54. Landman, B.A., Warfield, S.K.: MICCAI 2012: Workshop on Multi-atlas Labeling. éditeur non identifié (2019)
55. Rohlfing, T.: Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable. IEEE Trans. Med. Imaging 31(2), 153–163 (2011)
56. Kennedy, D.N., Haselgrove, C., Hodge, S.M., Rane, P.S., Makris, N., Frazier, J.A.: Candishare: A Resource for Pediatric Neuroimaging data. Springer, Berlin (2012)
57. Jun, M., Yixin, W., Xingle, A., Cheng, G., Ziqi, Y., Jianan, C., et al.: Towards efficient covid-19 CT annotation: A benchmark for lung and infection segmentation. arXiv:200412537 (2020)
58. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, pp. 234–241. Springer, Berlin (2015)
59. Sorensen, T.A.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. Biol. Skar. 5, 1–34 (1948)
60. Rockafellar, R.T., Wets, R.J.B.: Variational Analysis, vol. 317. Springer Science & Business Media, Berlin (2009)
61. Birsan, T., Tiba, D.: One hundred years since the introduction of the set distance by dimitrie pompeiu. In: IFIP Conference on System Modeling and Optimization, pp. 35–39. Springer, Heidelberg (2005)
62. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., et al.: beta-vae: Learning basic visual concepts with a constrained variational framework. Iclr 2(5), 6 (2017)

How to cite this article: Bdair, T., Wiestler, B., Navab, N., Albarqouni, S.: ROAM: Random layer mixup for semi-supervised learning in medical images. IET Image Process. 16, 2593–2608 (2022). https://doi.org/10.1049/ipr2.12511