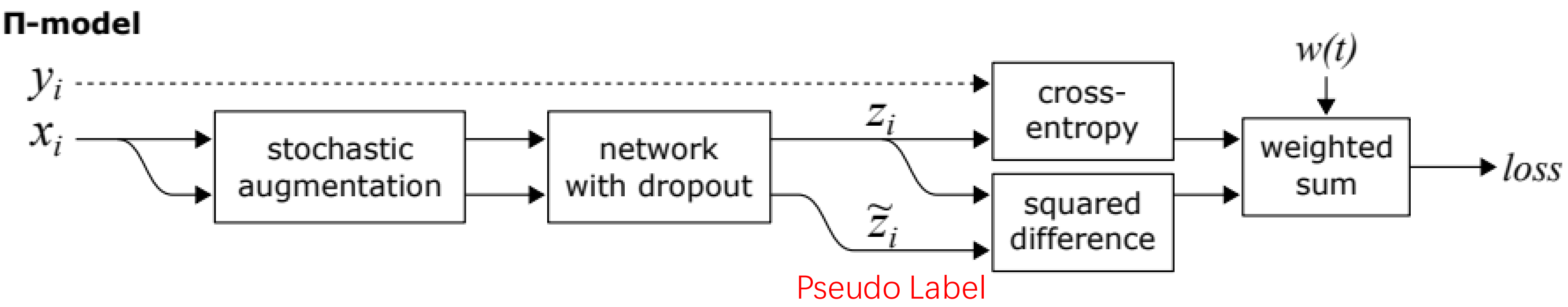
Semi-supervised Medical image segmentation

Huansen Chen

Background

- Medical image segmentation
  - MRI, CT, UT…
  - UNet, …
- Semi-supervised
  - **Smothness assumption:** for two input points x1, x2, that are close by in the input space, the corresponding labels y1, y2 should be the same.
  - **Consistency regularization:** a classifier should output the same class distribution for an unlabeled example even after it has been augmented.
- Other
  - **Ensemble method:** an ensemble of multiple neural networks generally yields better predictions than a single network in the ensemble

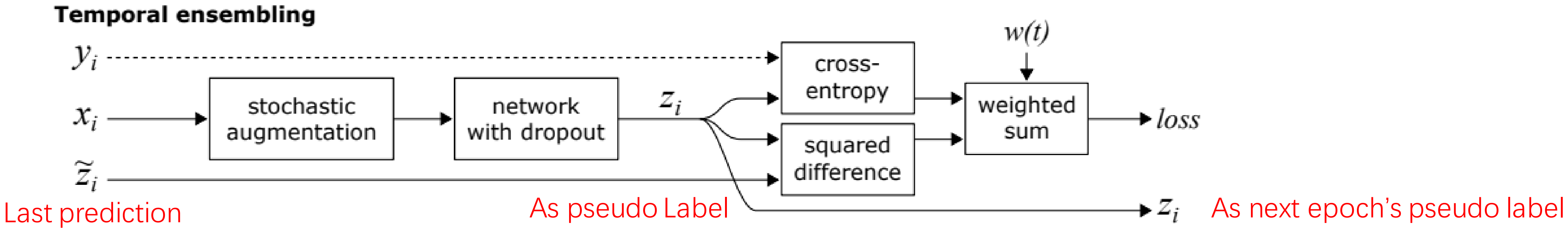
Pi-model



Algorithm 1 Π-model pseudocode.

Require: x_i = training stimuli
Require: L = set of training input indices with known labels
Require: y_i = labels for labeled inputs $i \in L$
Require: $w(t)$ = unsupervised weight ramp-up function
Require: $f_\theta(x)$ = stochastic neural network with trainable parameters θ
Require: $g(x)$ = stochastic input augmentation function
  for t in $[1, num\_epochs]$ **do**
    for each minibatch B **do**
      $z_{i \in B} \leftarrow f_\theta(g(x_{i \in B}))$ ▷ evaluate network outputs for augmented inputs
      $\tilde{z}_{i \in B} \leftarrow f_\theta(g(x_{i \in B}))$ ▷ again, with different dropout and augmentation
      $loss \leftarrow -\frac{1}{|B|} \sum_{i \in (B \cap L)} \log z_i[y_i]$ ▷ supervised loss component
          $+ w(t) \frac{1}{C|B|} \sum_{i \in B} ||z_i - \tilde{z}_i||^2$ ▷ unsupervised loss component
      update θ using, e.g., ADAM ▷ update network parameters
    end for
  end for
  return θ

Temporal ensembling



Algorithm 2 Temporal ensembling pseudocode. Note that the updates of Z and $\tilde{z}$ could equally well be done inside the minibatch loop; in this pseudocode they occur between epochs for clarity.

Require: x_i = training stimuli
Require: L = set of training input indices with known labels
Require: y_i = labels for labeled inputs $i \in L$
Require: α = ensembling momentum, $0 \leq \alpha < 1$
Require: $w(t)$ = unsupervised weight ramp-up function
Require: $f_\theta(x)$ = stochastic neural network with trainable parameters θ
Require: $g(x)$ = stochastic input augmentation function

```
Z ← 0_[N×C]                                      ▷ initialize ensemble predictions
z̃ ← 0_[N×C]                                      ▷ initialize target vectors
for t in [1, num_epochs] do
  for each minibatch B do
    z_{i∈B} ← f_θ(g(x_{i∈B}, t))                 ▷ evaluate network outputs for augmented inputs
    loss ← − 1/|B| Σ_{i∈(B∩L)} log z_i[y_i]      ▷ supervised loss component
           + w(t) 1/(C|B|) Σ_{i∈B} ||z_i − z̃_i||²  ▷ unsupervised loss component
    update θ using, e.g., ADAM                   ▷ update network parameters
  end for
  Z ← αZ + (1 − α)z                              ▷ accumulate ensemble predictions
  z̃ ← Z/(1 − α^t)                                ▷ construct target vectors by bias correction
end for
return θ
```

$$Z \leftarrow \mathbf{0}_{[N\times C]},\quad \tilde{z} \leftarrow \mathbf{0}_{[N\times C]}$$

$$loss \leftarrow -\frac{1}{|B|}\sum_{i\in(B\cap L)}\log z_i[y_i] + w(t)\frac{1}{C|B|}\sum_{i\in B}\|z_i-\tilde{z}_i\|^2$$

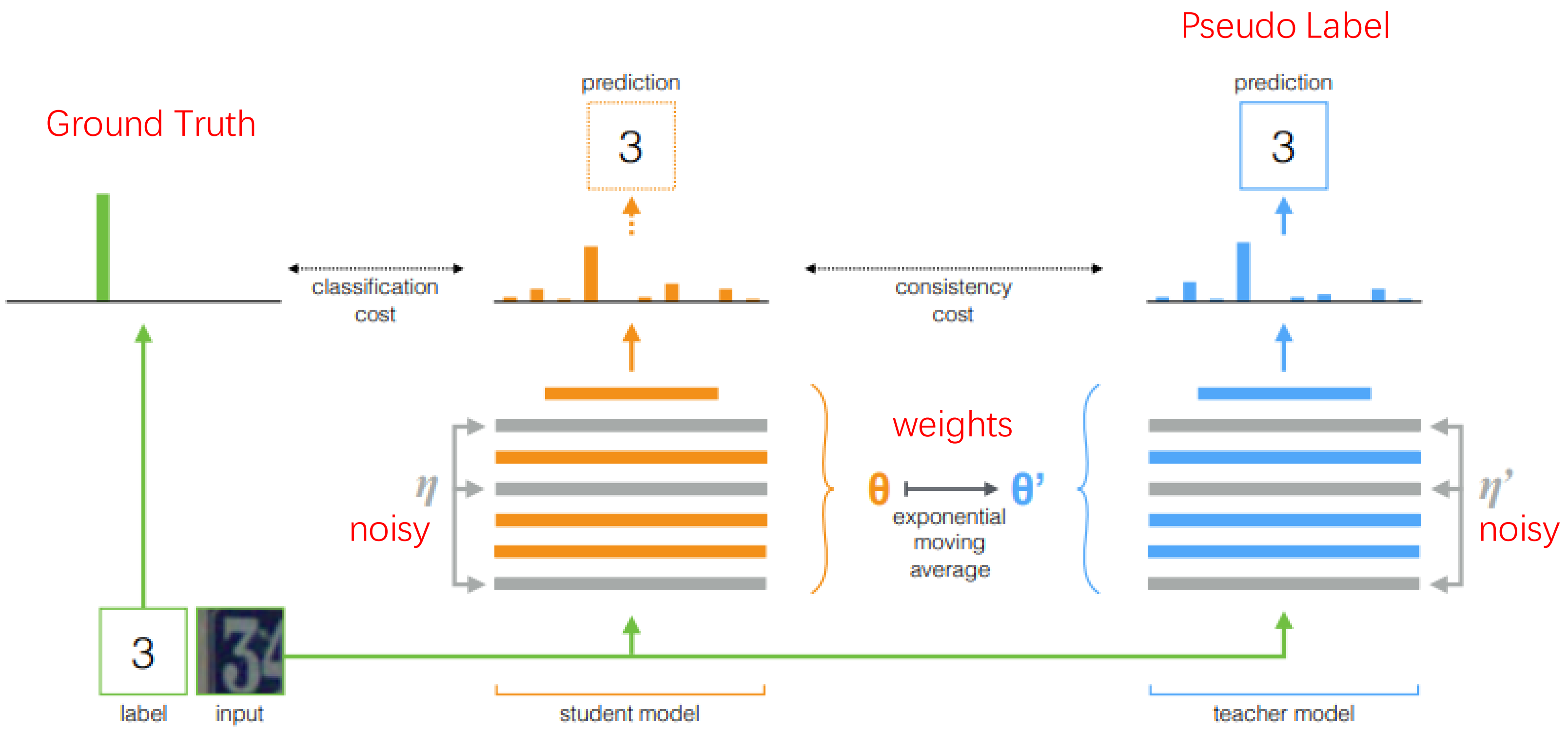
$$Z \leftarrow \alpha Z + (1-\alpha)z,\quad \tilde{z} \leftarrow Z/(1-\alpha^t)$$

1. Faster and downstream easily
2. Less noisy than with Pi-model

Mean teacher

- Since each target (of temporal ensembling) is updated **only once per epoch**, the learned information is incorporated into the training process at a slow pace
- To overcome the limitations of Temporal Ensembling, we propose averaging **model weights** instead of **predictions**
- Since the teacher model is an average of **consecutive student models**, we call this the Mean Teacher method

Mean teacher



1. **Teacher model weights** are updated as an **exponential moving average** of the student weights.
2. The softmax output of the student model is compared with the one-hot label using classification cost and with the **teacher output** using **consistency cost**.

(UA-MT) Uncertainty-Aware Self-ensembling Model for Semi-supervised 3D Left Atrium Segmentation

MICCAI 2019

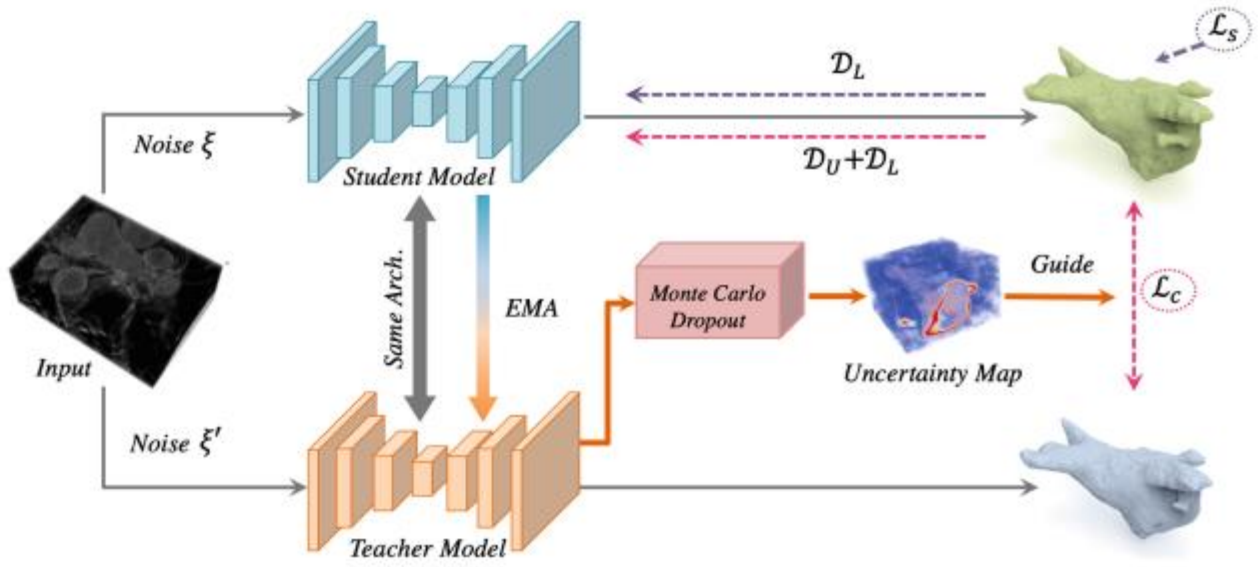
Motivation

1. Following the same spirit of mean teacher
2. The predicted target from the teacher model may be **unreliable** and **noisy** → **Uncertainty**
3. By **exploiting** the **uncertainty** information of the teacher model, the student model can learns from the **meaningful** and **reliable** targets

Method

1. The teacher model **estimates** the uncertainty of each target prediction with **Monte Carlo sampling (dropout)**
2. With the guidance of the estimated uncertainty, we **filter out** the unreliable predictions and preserve only the reliable ones (**low uncertainty**)

Framework



$$\min_{\theta} \sum_{i=1}^{N} \mathcal{L}_s(f(x_i;\theta), y_i) + \lambda \sum_{i=1}^{N+M} \mathcal{L}_c(f(x_i;\theta', \xi'), f(x_i;\theta, \xi))$$

D_L: label data **D_u**: unlabel data
L_s: supervised loss **L_c**: consistent loss
Monte Carlo Dropout: perform T stochastic forward passes on the teacher model under random dropout and input Gaussian noise for each input volume
Uncertainty map: the predictive entropy can be summarized as:

$$\mu_c = \frac{1}{T}\sum_t \mathbf{p}_t^c \quad \text{and} \quad u = -\sum_c \mu_c \log \mu_c,$$

p_t^c is the probability of the c-th class in the t-th time prediction
Consistency Loss:

$$\mathcal{L}_c(f', f) = \frac{\sum_v \mathbb{I}(u_v < H)\, \|f'_v - f_v\|^2}{\sum_v \mathbb{I}(u_v < H)},$$

U_v is the estimated uncertainty U at the v-th voxel; and H is a threshold to select the most certain targets

Experiment-Visualization

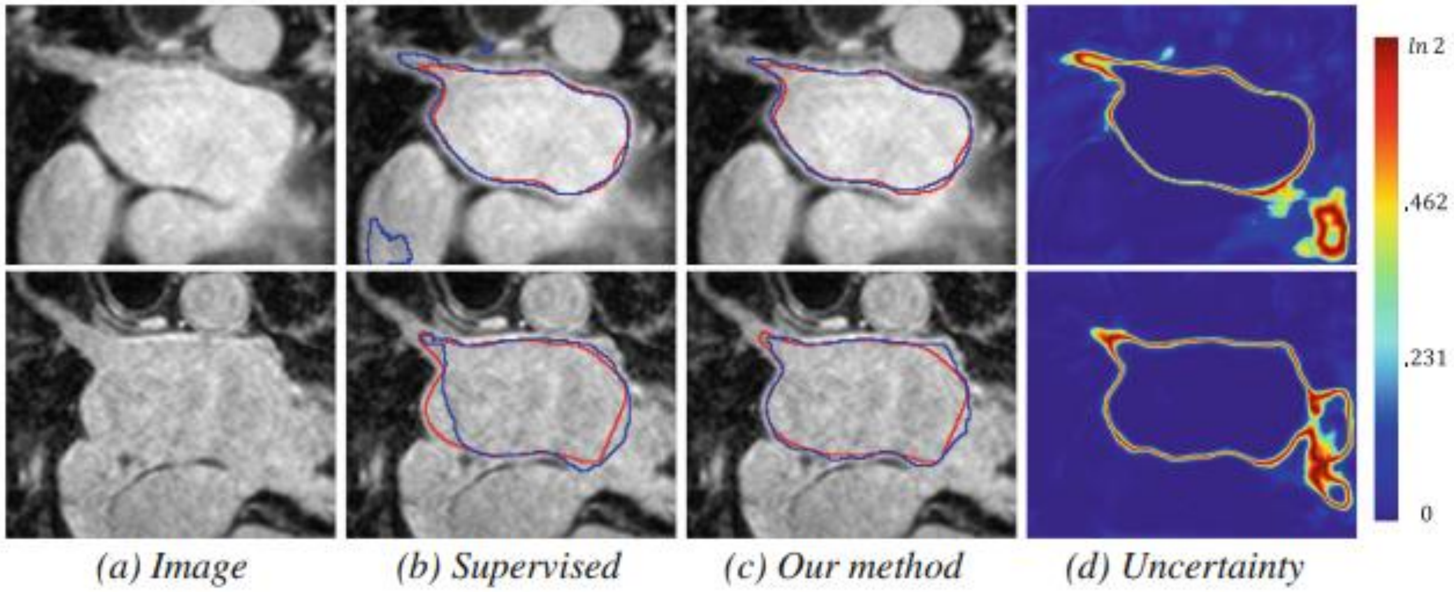


Fig. 2. Visualization of the segmentations by different methods and the uncertainty. Blue and red colors show the predictions and ground truths, respectively. (Color figure online)

Experiment-Comparison

Table 1. Comparison between our method and various methods.

Method	# scans used		Metrics			
	Labeled	Unlabeled	Dice [%]	Jaccard [%]	ASD [voxel]	95HD [voxel]
Vanilla V-Net	16	0	84.13	73.26	4.75	17.93
Bayesian V-Net	16	0	86.03	76.06	3.51	14.26
Vanilla V-Net	80	0	90.25	82.40	1.91	8.29
Bayesian V-Net	80	0	91.14	83.82	1.52	5.75
Self-training [1]	16	64	86.92	77.28	2.21	9.19
DAN [18]	16	64	87.52	78.29	2.42	9.01
ASDNet [12]	16	64	87.90	78.85	**2.08**	9.24
TCSE [10]	16	64	88.15	79.20	2.44	9.57
UA-MT-UN (ours)	16	64	88.83	80.13	3.12	10.04
UA-MT (ours)	16	64	**88.88**	**80.21**	2.26	**7.32**

Experiment-Quantitative analysis

Table 2. Quantitative analysis of our method.

Method	# scans used		Metrics			
	Labeled	Unlabeled	Dice [%]	Jaccard [%]	ASD [voxel]	95HD [voxel]
MT	16	64	88.23	79.29	2.73	10.64
MT-Dice [5]	16	64	88.32	79.37	2.76	10.50
Our UA-MT	16	64	88.88	80.21	2.26	7.32
Bayesian V-Net	8	0	79.99	68.12	5.48	21.11
Our UA-MT	8	72	84.25	73.48	3.36	13.84
Bayesian V-Net	24	0	88.52	79.70	2.60	10.45
Our UA-MT	24	56	90.16	82.18	2.73	8.90

(SASSNet) Shape-Aware Semi-supervised 3D Semantic Segmentation for Medical Images

MICCAI 2020

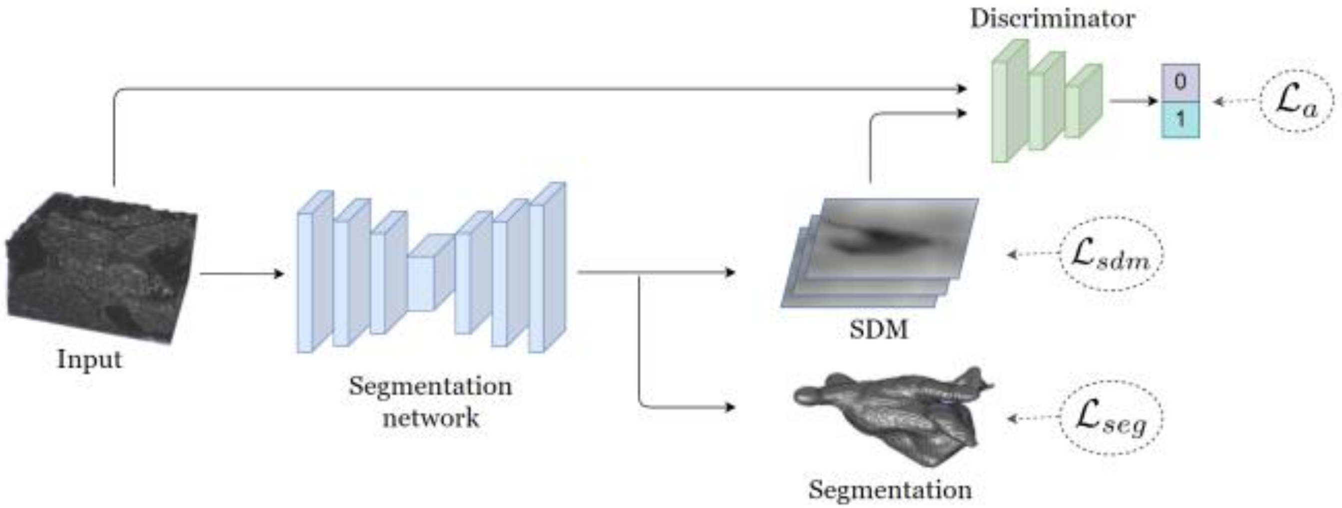
Motivation

1. Some semi-supervised methods lack explicit modeling of the **geometric prior** of semantic objects, often leading to **poor object coverage** and/or boundary prediction
2. Some priors typically assumes **properly aligned input images**, which is **difficult** to achieve **in practice** for objects with large variation in pose or shape

Contribution

1. propose a novel shape-aware semi-supervised segmentation approach by enforcing **geometric constraints** on labeled and unlabeled data
2. develop a **multi-task loss** on segmentation and SDM (signed distance map) predictions, and impose global consistency in object shapes through adversarial learning

Framework



optimization function:

$$\min_{\theta} \max_{\zeta} \mathcal{V}(\theta, \zeta) = \mathcal{L}_s(\theta) + \beta \mathcal{L}_a(\theta, \zeta)$$

$$\mathcal{L}_s(\theta) = \mathcal{L}_{seg} + \alpha \mathcal{L}_{sdm}$$

Definition [edit]

If Ω is a subset of a metric space, X, with metric, d, then the *signed distance function*, f, is defined by

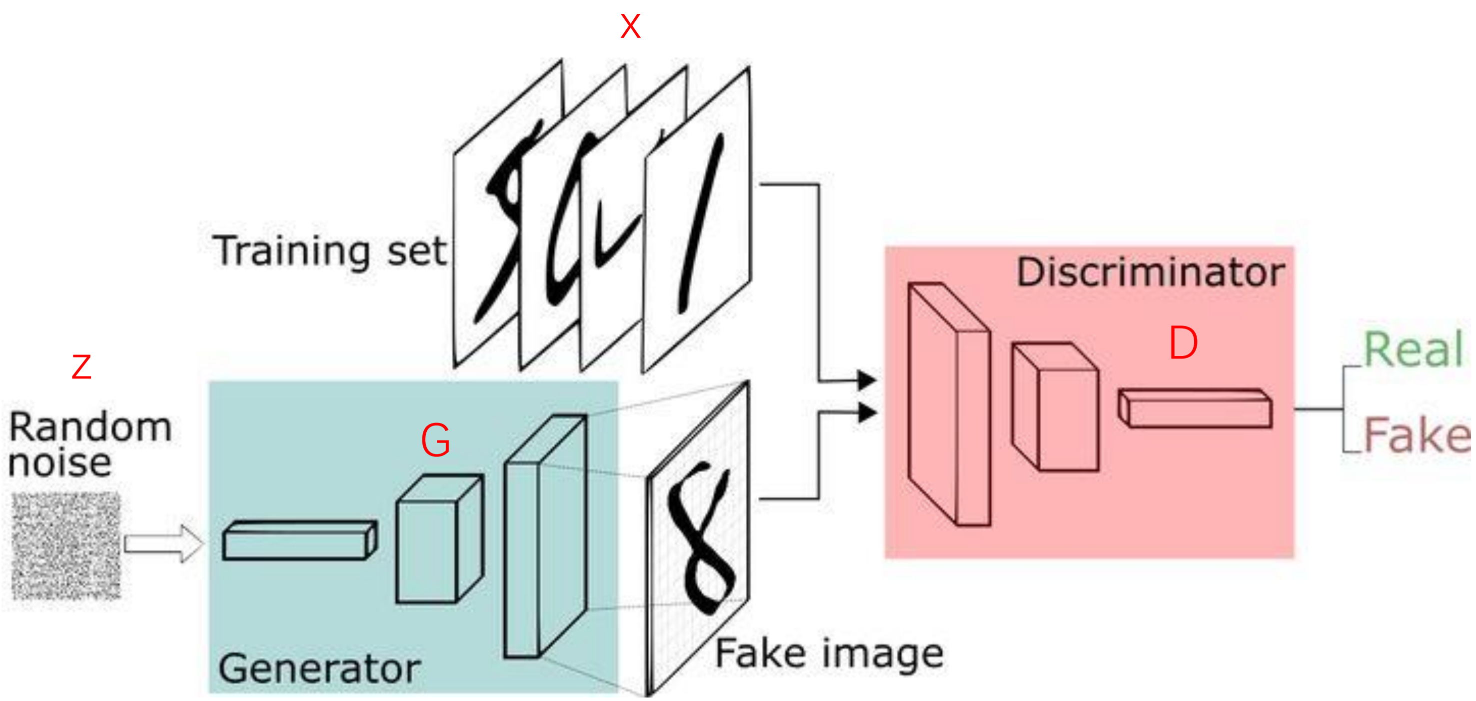
$$f(x) = \begin{cases} d(x, \partial\Omega) & \text{if } x \in \Omega \\ -d(x, \partial\Omega) & \text{if } x \in \Omega^c \end{cases}$$

where $\partial\Omega$ denotes the boundary of Ω. For any $x \in X$,

$$d(x, \partial\Omega) := \inf_{y \in \partial\Omega} d(x, y)$$

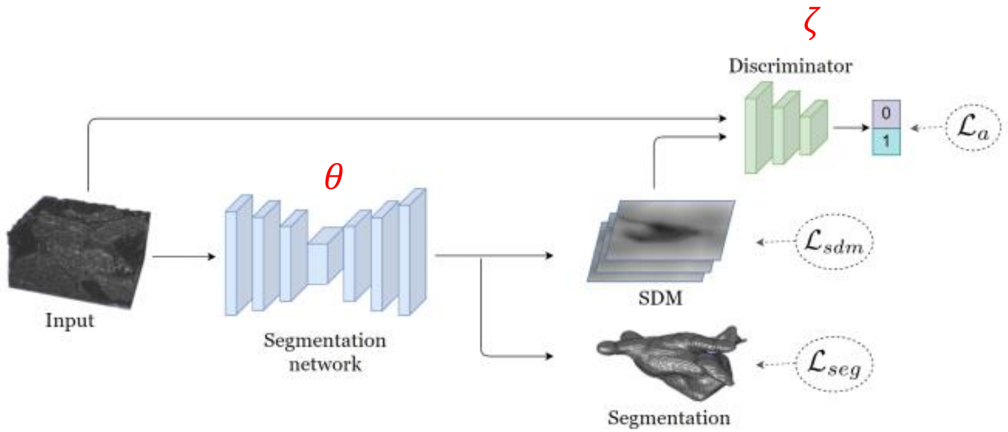
where $\inf$ denotes the infimum.

Generative Adversarial Network (GAN)



$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Method



optimization function:

$$\min_{\theta}\max_{\zeta}\mathcal{V}(\theta,\zeta)=\mathcal{L}_s(\theta)+\beta\mathcal{L}_a(\theta,\zeta)$$

Supervised loss:

$$\mathcal{L}_s(\theta)=\mathcal{L}_{seg}+\alpha\mathcal{L}_{sdm}$$

$$\mathcal{L}_{seg}=\frac{1}{N}\sum_{i=1}^{N}l_{dice}(f_{seg}(\mathbf{X}_i;\theta),\mathbf{Y}_i);\quad \mathcal{L}_{sdm}=\frac{1}{N}\sum_{i=1}^{N}l_{mse}(f_{sdm}(\mathbf{X}_i;\theta),\mathbf{Z}_i)$$

Zi are the GT of SDMs derived from Yi

Adversarial loss:

From labeled set (real) From unlabeled set (fake)

$$\mathcal{L}_a(\theta,\zeta)=\frac{1}{N}\sum_{n=1}^{N}\log D(\mathbf{X}_n,\mathbf{S}_n;\zeta)+\frac{1}{M}\sum_{m=N+1}^{N+M}\log\left(1-D(\mathbf{X}_m,\mathbf{S}_m;\zeta)\right)$$

Shape-aware representation

where $\mathbf{S}_n=f_{sdm}(\mathbf{X}_n;\theta)$ and $\mathbf{S}_m=f_{sdm}(\mathbf{X}_m;\theta)$ are the predicted SDMs.

Experiment-Visualization

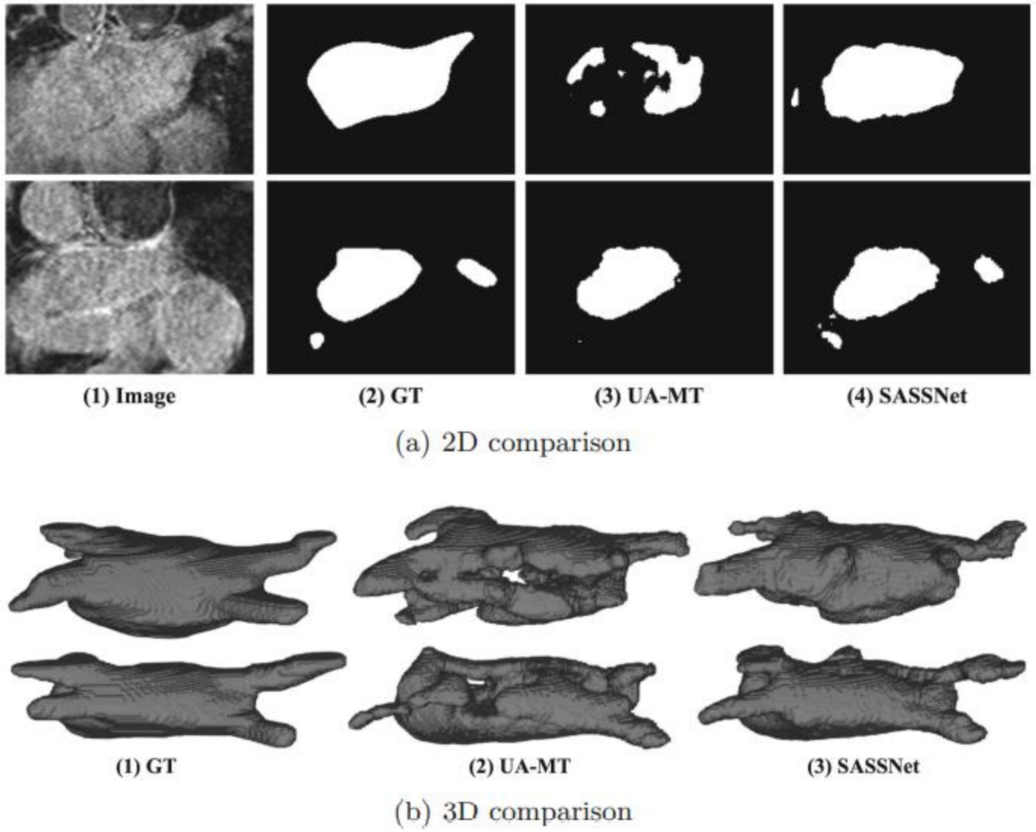


Fig. 2. 2D and 3D Visualization of the segmentations by UA-MT [18] and our method, where GT denotes groundtruth segmetnation.

Experiment-Comparison

Table 1. Quantitative comparisons of semi-supervised segmentation models on the LA dataset. All models use the V-Net as backbone network. Results on two different data partition settings show that our SASSNet outperforms the state-of-the-art results consistently.

Method	# Scans used		Metrics			
	Labeled	Unlabeled	Dice [%]	Jaccard [%]	ASD [voxel]	95HD [voxel]
V-Net	80	0	91.14	83.82	1.52	5.75
V-Net	16	0	86.03	76.06	3.51	14.26
DAP [20]	16	64	87.89	78.72	2.74	9.29
ASDNet [11]	16	64	87.90	78.85	**2.08**	9.24
TCSE [9]	16	64	88.15	79.20	2.44	9.57
UA-MT [18]	16	64	88.88	80.21	2.26	7.32
UA-MT (+NMS)	16	64	89.11	80.62	2.21	**7.30**
SASSNet (ours)	16	64	89.27	80.82	3.13	8.83
SASSNet (+NMS)	16	64	**89.54**	**81.24**	2.20	8.24
V-Net	8	0	79.99	68.12	5.48	21.11
DAP [20]	8	72	81.89	71.23	3.80	15.81
UA-MT [18]	8	72	84.25	73.48	3.36	13.84
UA-MT(+NMS)	8	72	84.57	73.96	2.90	12.51
SASSNet(ours)	8	72	86.81	76.92	3.94	12.54
SASSNet(+NMS)	8	72	**87.32**	**77.72**	**2.55**	**9.62**

Experiment-Ablative study

Table 2. Effectiveness of our proposed modules on the LA dataset. All the models use the same V-Net as the backbone, and we conduct an ablative study to show the contribution of each component module.

Method	# Scans used		Metrics				Cost
	Labeled	Unlabeled	Dice [%]	Jaccard [%]	ASD [voxel]	95HD [voxel]	Params [M]
V-Net	8	0	79.99	68.12	5.48	21.11	187.7
V-Net +SDM	8	0	81.12	69.75	6.93	25.58	187.9
V-Net +SDM +GAN	8	72	86.81	76.92	3.94	12.54	249.7
UA-MT [18]	8	72	84.25	73.48	3.36	13.84	375.5
V-Net +SDM +MT	8	72	84.97	74.14	6.12	22.20	375.8

Thanks