Supplementary Material: MCSC

Qianying Liu[1]
2665227L@student.gla.ac.uk
Xiao Gu[2]
xiao.gu17@imperial.ac.uk
Paul Henderson[1]
paul.henderson@glasgow.ac.uk
Fani Deligianni[1]
fani.deligianni@glasgow.ac.uk

[1] School of Computing Science
University of Glasgow
Glasgow, UK

[2] Department of Computing
Imperial College London
London, UK

S1 Pseudocode

Algorithm 1: Loss calculation for one minibatch with MCSC.

Input: Batch of images $X = X^l \cup X^u$ including labelled images and unlabelled images, ground-truth Y^l for labelled images, temperature constant τ, and N the number of feature scales.

Output: Total losses $\mathcal{L}_c$ for CNN and $\mathcal{L}_t$ for Transformer.

$P_*^{u/l} = \text{softmax}\{C_*(E_*(X^{u/l}))\}$ // Compute class probability maps on unlabelled data X^u and labelled data X^l

$Y_*^u = \text{argmax}(P_*^u)$ // Compute pseudo one-hot label map on unlabelled data X^u

Supervised Supervision

$\mathcal{L}_{sup(*)} = \mathcal{L}_{dice}(P_*^l, Y_*^l) + \mathcal{L}_{ce}(P_*^l, Y_*^l)$

Cross Pseudo Supervision

$\mathcal{L}_{cps(c)} = \mathcal{L}_{dice}(P_c^u, Y_t^u)$

$\mathcal{L}_{cps(t)} = \mathcal{L}_{dice}(P_t^u, Y_c^u)$

Multi-Scale Cross Supervised Contrastive Learning

for $n = 1 \ldots N$ **do**

  $F_* = H_*(E_*(X))$, $F = \text{concat}(F_c, F_t)$ //Get a feature batch F from layer n of extractors followed by projectors

  $M = (h/h')^2$, $\{A^m\}_{m=1}^M = F$ // divide F into M groups of patches A

  Define: $\mathcal{L}_{bcl}(A) = -\frac{1}{|A|}\sum_{a_i \in A} \frac{1}{|A_y|-1} \sum_{p \in A_y \setminus \{i\}} \log \frac{\exp(a_i \cdot a_p/\tau)}{\sum_{j \in Y_A} \frac{1}{|A_j|} \sum_{a_k \in A_j} \exp(a_i \cdot a_k/\tau)}$ //a_i is the i^{th} feature sample, $A_y \subseteq A$ is the subset of features associated with class y where $Y_{t/c}$ defines the class of $F_{c/t}$, and Y_A is the set of all classes present in A

  $\mathcal{L}_{cl\_n} = \frac{1}{|M|}\sum_{m=1}^M \mathcal{L}_{bcl}(A^m)$ // Average over M groups to get the loss of F

end

$\mathcal{L}_{cl} = (\mathcal{L}_{cl\_1} + \ldots + \mathcal{L}_{cl\_N})$ // Sum each scale balanced contrastive loss

$\mathcal{L}_* = \mathcal{L}_{sup(*)} + w_{cps}\mathcal{L}_{cps(*)} + w_{cl}\mathcal{L}_{cl}$

Return: $\mathcal{L}_c, \mathcal{L}_t$

Algorithm 1 gives the pseudocode for MCSC processing a single mini-batch of data.

S2 Implementation details

We implemented our method in PyTorch. We used simple data augmentations to reduce overfitting: random cropping with a 224×224 patch, random flipping and rotations. All methods were trained till validation-set convergence (which was by 40,000 iterations). We selected the best checkpoint for evaluation based on validation set performance. Our method was trained using AdamW [3] with a weight decay of 5×10^{-4}. We utilized the poly learning rate schedule, initialized at 5×10^{-4} for CNN and 1×10^{-4} for Transformer. The batch sizes were 4 and 10 respectively, with half labeled and half unlabeled images. For our MCSC module, each projector H_* has two linear layers, where the first linear layer changes the dimension of feature map to 256 channels; the last layer has 128 channels and shares its parameters between the two models. In Eq.(2), temperature $\tau = 0.1$. We use multi-scale feature maps from three layers of E_*, with sizes of 256×256, 56×56, and 28×28 respectively, and the size h' of a patch was set to 19, 28 and 14 accordingly. All experiments were run on one (for ACDC) or two (for Synapse) RTX 3090 GPUs.

S3 Full results on ACDC and Synapse

Table S1 summarizes ACDC segmentation results of our MCSC and all baselines on 7 and 3 labelled cases, and results of our MCSC, UNet-LS and CTS on 1 case. Segmentation visualizations from our method, LS and CST trained on 7 cases on ACDC are shown in Figure S1.

Table S2 shows the segmentation results of all methods on Synapse dataset under two settings (4 and 2 labelled cases). Figure S2 shows segmentation visualizations for UNet with limited supervision, CPS, CTS and our method MCSC, on 4 cases of Synapse.

Labeled	**Methods**	DSC↑	HD↓	Aorta	Gallb	Kid_L	Kid_R	Liver	Pancr	Spleen	Stom
18 cases(100 %)	UNet-FS	75.6	42.3	88.8	56.1	78.9	72.6	91.9	55.8	85.8	74.7
	nnFormer [10]	86.6	10.6	92.0	70.2	86.6	86.3	96.8	83.4	90.5	86.8
4 cases(20 %)	UNet-LS	47.2	122.3	67.6	29.7	47.2	50.7	79.1	25.2	56.8	21.5
	UAMT	51.9	69.3	75.3	33.4	55.3	40.8	82.6	27.5	55.9	44.7
	ICT	57.5	79.3	74.2	36.6	58.3	51.7	86.7	34.7	66.2	51.6
	CCT	51.4	102.9	71.8	31.2	52.0	50.1	83.0	32.5	65.5	25.2
	CPS	57.9	62.6	75.6	41.4	60.1	53.0	88.2	26.2	69.6	48.9
	CTS	64.0	56.4	**79.9**	38.9	66.3	63.5	86.1	41.9	75.3	60.4
	MCSC (Ours)	**68.5**	**24.8**	76.3	**44.4**	**73.4**	**72.3**	**91.8**	**46.9**	**79.9**	**62.9**
2 cases(10 %)	UNet-LS	45.2	55.6	66.4	**27.2**	46.0	48.0	82.6	18.2	39.9	33.4
	UAMT	49.5	62.6	71.3	21.1	62.6	51.4	79.3	22.8	58.2	29.0
	ICT	49.0	59.9	68.9	19.9	52.5	52.2	83.7	25.4	53.2	36.0
	CCT	46.9	58.2	66.0	26.6	53.4	41.0	82.9	21.2	48.7	35.6
	CPS	48.8	65.6	70.9	21.3	58.0	45.1	80.7	23.5	58.0	32.7
	CTS	52.0	63.7	73.2	12.7	67.2	64.7	82.9	31.7	40.9	42.4
	MCSC (Ours)	**61.1**	**32.6**	**73.9**	26.4	**69.9**	**72.7**	**90.0**	**33.2**	**79.4**	**43.0**

Best is reported as bold, Second Best is underlined.

Table S2: Comparison with different models on Synapse. The performance is reported by DSC (%) and HD (%), as well as the DSC value of each types of organs.

Labeled	Methods	Mean		Myo		LV		RV	
		DSC↑	HD↓	DSC↑	HD↓	DSC↑	HD↓	DSC↑	HD↓
70 cases (100%)	UNet-FS	91.7	4.0	89.0	5.0	94.6	5.9	91.4	1.2
	BATFormer [4]	92.8	8.0	90.26	6.8	96.30	5.9	91.97	11.3
7 cases (10%)	UNet-LS	75.9	10.8	78.2	8.6	85.5	13.0	63.9	10.7
	MT [4]	80.9	11.5	79.1	7.7	86.1	13.4	77.6	13.3
	DCT [4]	80.4	13.8	79.3	10.7	87.0	15.5	75.0	15.3
	UAMT [4]	81.1	11.2	80.1	13.7	87.1	18.1	77.6	14.7
	ICT [4]	82.4	7.2	81.5	7.8	87.6	10.6	78.2	3.2
	CCT [4]	84.0	6.6	82.3	5.4	88.6	9.4	81.0	5.1
	CPS [4]	85.0	6.6	82.9	6.6	88.0	10.8	84.2	2.3
	CTS [4]	86.4	8.6	84.4	6.9	90.1	11.2	84.8	7.8
	MCSC (Ours)	**89.4**	**2.3**	**87.6**	**1.1**	**93.6**	**3.5**	**87.1**	**2.1**
3 cases (5%)	UNet-LS	51.2	31.2	54.8	24.4	61.8	24.3	37.0	44.4
	MT [4]	56.6	34.5	58.6	23.1	70.9	26.3	40.3	53.9
	DCT [4]	58.2	26.4	61.7	20.3	71.7	27.3	41.3	31.7
	UAMT [4]	61.0	25.8	61.5	19.3	70.7	22.6	50.8	35.4
	ICT [4]	58.1	22.8	62.0	20.4	67.3	24.1	44.8	23.8
	CCT [4]	58.6	27.9	64.7	22.4	70.4	27.1	40.8	34.2
	CPS [4]	60.3	25.5	65.2	18.3	72.0	22.2	43.8	35.8
	CTS [4]	65.6	16.2	62.8	11.5	76.3	15.7	57.7	21.4
	MCSC (Ours)	**73.6**	**10.5**	**70.0**	**8.8**	**79.2**	**14.9**	**71.7**	**7.8**
1 case	UNet-LS	26.4	60.1	26.3	51.2	28.3	52.0	24.6	77.0
	CTS [4]	46.8	36.3	55.1	**5.5**	64.8	**4.1**	20.5	99.4
	MCSC (Ours)	**58.6**	**31.2**	**64.2**	13.3	**78.1**	12.2	**33.5**	**68.1**

Best is reported as bold, Second Best is underlined.

Table S1: Segmentation results on DSC(%) and HD(mm) of our method and baselines on ACDC, across different numbers of labelled cases. Bold is the best result, and underlined 2nd-best, for each number of cases.

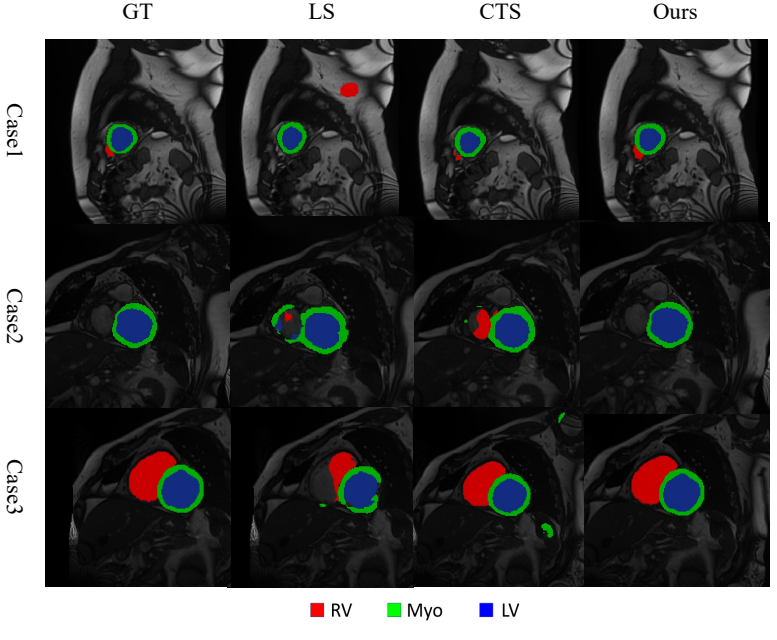

Figure S1: Segmentation visualizations from our method, LS and CST trained on 7 labelled cases on ACDC.

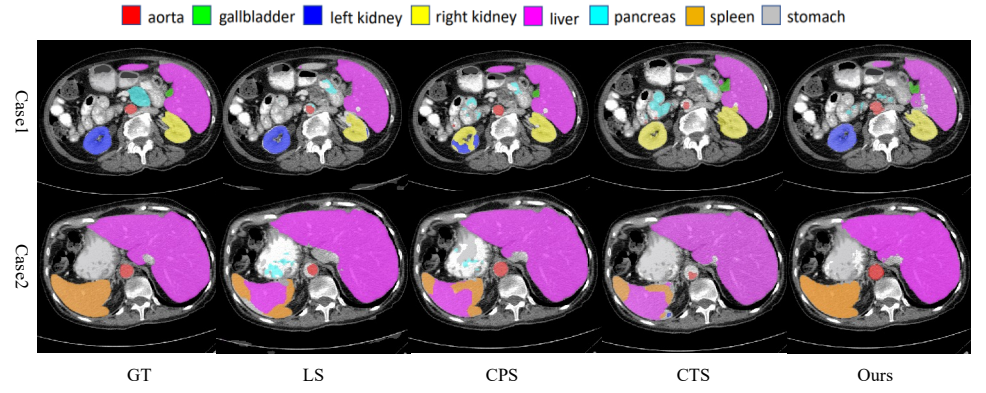

Figure S2: Segmentation visualizations from our method, LS, CPS and CST trained on 4 labelled cases on Synapse.

S4 Visualization of features with contrastive learning

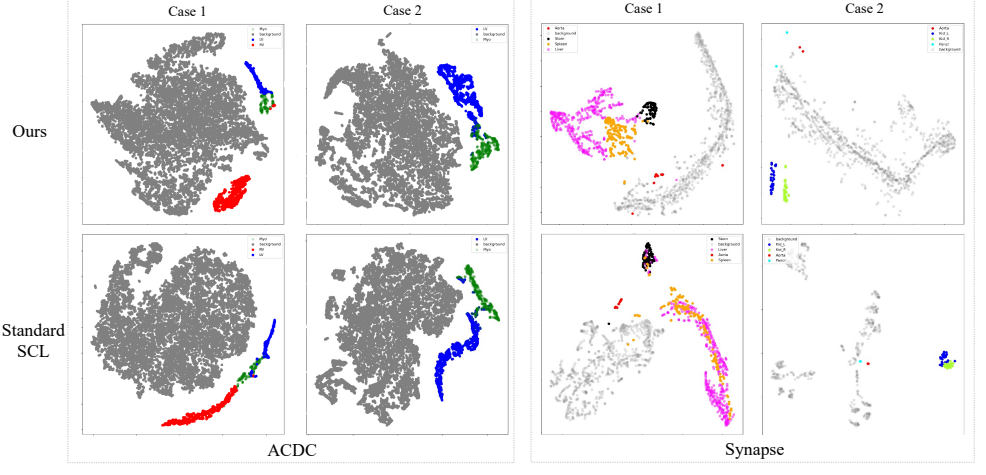

Figure S3: Visualization of embedding features from our method and standard SCL after applying t-SNE on test data of ACDC and Synapse respectively. Each case represents one slice from a different patient.

Figure S3 shows visualizations of embedding features applying t-SNE on a single slice of two cases from the test data of ACDC and Synapse respectively. The models are trained with 7 cases (ACDC) and 4 cases (Synapse). Different colors represent different classes. Features are taken from the feature map after the projector with scale of 256×256, and each point in the figure is the embedding of one pixel. The standard SCL is the second row of Table 3 in the main text (SCL+DB). For ACDC, the left case 1 shows our method better separates RV from the other two foreground classes, and reduces the overlap between LV and Myo. For the case 2, the foreground clusters of ours are tighter. A more clear and consistent effect can be seen on Synapse. For the case on the left, our method makes the liver, spleen and stomach

much better separated than standard SCL. A similar situation also occurs with the left and right kidneys for the case 2. Overall, through cross labelling, averaging the contribution of each class in SCL, and contrasting multi-scale feature maps, our method obtains a better embedding representations for segmentation, where features within the same class are pulled closer and features for different class are spread farther apart.

References

[1] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

[2] Xian Lin, Li Yu, Kwang-Ting Cheng, and Zengqiang Yan. Batformer: Towards boundary-aware lightweight transformer for efficient medical image segmentation. *IEEE Journal of Biomedical and Health Informatics*, 2023.

[3] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[4] Xiangde Luo, Minhao Hu, Tao Song, Guotai Wang, and Shaoting Zhang. Semi-supervised medical image segmentation via cross teaching between cnn and transformer. In *International Conference on Medical Imaging with Deep Learning*, pages 820–833. PMLR, 2022.

[5] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[6] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, pages 135–152, 2018.

[7] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[8] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022.

[9] Lequan Yu, Shujun Wang, Xiaomeng Li, Chi-Wing Fu, and Pheng-Ann Heng. Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22*, pages 605–613. Springer, 2019.

[10] Hong-Yu Zhou, Jiansen Guo, Yinghao Zhang, Lequan Yu, Liansheng Wang, and Yizhou Yu. nnformer: Interleaved transformer for volumetric segmentation. *arXiv preprint arXiv:2109.03201*, 2021.