Structure-aware Semantic Discrepancy and Consistency for 3D Medical Image Self-supervised Learning

Supplementary Material

A. Overview

In Appendix B, we introduce the data splits and collections of pre-training and downstream datasets. In Appendix C, we introduce details of the pre-training and fine-tuning process. In Appendix D, we give brief descriptions of some algorithms. In Appendix E, we discuss the performance of SSL and mSSL methods on the CC-CCII dataset. In Appendix F, we visualize the results of downstream tasks and patch-to-patch correspondence after pre-training.

B. Data collections and splits

We follow previous open-source data splits and split datasets without public splits.

Pretraining datasets. For the CT data, we use the 10k collection provided by the official implementation of the method [5]. For MRI data, we collect the OASIS dataset containing T1, T2, Flare, and MRA sequences and filter out images with the shortest side smaller than 64, resulting in a total of 6,605 volumes. For PET data, we utilize the whole-body full-dose volumes from UDPET as well as brain volumes from ADNI, resulting in a total of 3,519 volumes. The splits of data collection will be released upon publication.

Modality	Dataset	Samples
CT	BTCV	24
CT	TCIA Covid19	722
CT	FLARE23	4000
CT	HNSCC	1071
CT	LiDC	589
CT	LUNA16	843
CT	TotalSegmentor	1203
CT	STOIC 2021	2000
Total	-	10502
MRI	OASIS3	6605
PET	UDPET	1238
PET	ADNI	2281
Total	-	3519

Table 1. The data distribution of training datasets.

Downstream datasets. For downstream tasks, we follow previous data splits [3, 5] on BTCV, CC-CCII, BraTs 21, and BraTs 23. For other datasets, we split the data by a 2:8 ratio. In the CC-CCII dataset, instances are classified as novel coronavirus pneumonia (NCP), common pneumonia (CP), and normal controls (Normal). In the ADNI dataset, the split is performed at the subject level, where a single subject may have multiple imaging volumes. The data details are as shown in Tab. 2:

Modality	Dataset	Task	Samples
CT	BTCV	Segmentation	30
CT	MSD-Liver	Segmentation	131
CT	MSD-Lung	Segmentation	63
CT	MSD-Spleen	Segmentation	41
MRI	BraTs 21	Segmentation	1251
PET	AUTOPET	Segmentation	1014
CT	CC-CCII	Classification	4178
PET	ADNI	Classification	2876
MRI	BraTs 23	Image-to-Image translation	1470
PET	UDPET	Reconstruction	377

Table 2. Dataset details of downstream tasks.

C. Implementation details

For pre-training, we follow previous work [3–5] and use one A100 GPU for CT training in the 1k dataset. We use four A100 GPUs for the 10k CT dataset, 3k PET dataset, and 6k MRI dataset separately. We employ an AdamW optimizer with the momentum set to 0.9. All methods are trained with the input size $96 \times 96 \times 96$. For the size of the memory queue in $\mathcal{L}_g$, we adopt 90 and 200 for 1k CT data and other pre-training separately.

For fine-tuning on downstream tasks, we use four A100 GPUs for fine-tuning in AUTOPET and BraTs 23 while using one A100 GPU for other datasets. Specifically, for classification tasks, we employ Swin-B with MLP layers as the backbone. For the BraTs 21 and BraTs 23 tasks, the first layer of Swin-B is not initialized by the pre-trained model because of different input channels. For segmentation tasks, we adopt a sliding window strategy with an overlap ratio of 0.75. Following previous work partly, we set training epochs to 3000, 1000, 1000, 1000, 300, 400, 100, 300, and 300 for datasets corresponding to Tab. 2.

D. Brief descriptions of algorithms

D.1. Ground truth of patch correspondence: A toy sample

As shown in Fig. 1, the 2nd patch in $\mathcal{V}_i$ corresponds to the 4th patch in $\mathcal{V}_i^{'}$. In our implementation, the patch size is the same as the patch size of the tokenizer in Vision Transformer. The process of patch tokenizer and patch correspondence can be viewed in Fig. 2.

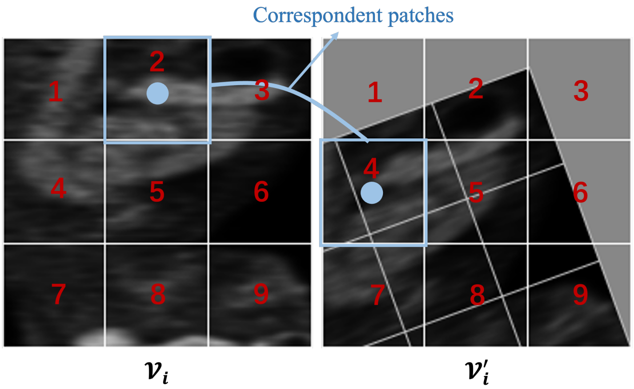

Figure 1. A toy sample to describe calculating ground truth of correspondent patches.

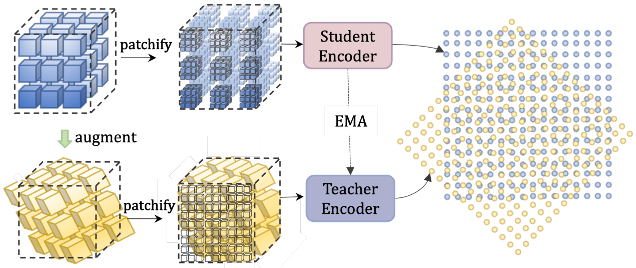

Figure 2. The visualization of patch-to-patch correspondence of different methods.

D.2. The dictionary of contrastive learning

The keys in the dictionary consist of two sets:

The first set is $\{q_j^* = \phi^* \circ \varphi^*(v_j) | v_j \in \mathcal{V}, j \neq i, j = 1, 2, ..., N\} \cup \{q_i^{*'} = \phi^* \circ \varphi^*(v_i^{'})\}$, encoded on-the-fly by the momentum-updated encoder $\phi^* \circ \varphi^*$, where ϕ^* and φ^* are the projection head and the encoder of the teacher model (e.g., Vision Transformer [1, 2]). q_j^* and $q_j^{*'}$ are the global features encoded by $\phi^* \circ \varphi^*$. A feature will be randomly selected from $\{q_j^* = \phi^* \circ \varphi^*(v_j) | v_j \in \mathcal{V}, j \neq i, j = 1, 2, ..., N\}$ to update the queue.

The second set is the feature set from the queue $\{y_k^* | k = 1, 2, ..., K\}$, which is updated by every training iteration, where K is the length of the queue.

By combining both sets, we obtain the dictionary $\{q_m^* | m = 1, 2, ..., K + N - 1\}$. The negative samples in the dictionary include both intra-subvolumes from the same volume and inter-subvolumes from different volumes. This negative sampling strategy increases the diversity of the samples.

D.3. Sinkhorn & dual softmax

We compare the results of different iteration numbers of Sinkhorn and the dual-softmax operator as follows:

Increasing the number of iterations for the Sinkhorn algorithm generally leads to improved segmentation performance for three tasks. The dual-softmax operator performs well across all three tasks, which is adopted for other modalities and tasks in this paper. The pseudocode of differentiable Sinkhorn in log domain is as follows, where we set $\lambda = 1$.

D.4. Soft regularization for patch-to-structure consistency

The optimal transport process aims to establish a patch-to-patch correspondence between patch sets across volumes. However, due to the lack of labels, it is unclear whether the patches we define for training correspond to a complete anatomical structure (e.g., a training patch may represent only a part of an organ). If the patch t_n from the volume $\mathcal{V}_i$ has multiple similar patches in volume $\mathcal{V}_i^{'}$, enforcing a strict patch-to-patch correspondence may distort this relationship.

To address this issue, we use the Sharpe ratio as a soft regularization factor for patch-to-patch loss. The Sharpe ratio is:

$$sr^n_{\mathcal{V}_i} = \frac{\max(\mathcal{D}_n) - \frac{1}{N}\sum_{m=1}^{N} \mathcal{D}_n}{\sigma_{\mathcal{D}_n}}, \tag{1}$$

where $\mathcal{D}_n$ represents the similarity vector of a patch feature t_n from $\mathcal{V}_i$ with the patch feature set from $\mathcal{V}_i^{'}$. Here, the similarity is measured using Cosine Similarity. The $sr^n_{\mathcal{V}_i}$ reflects the variability of the similarity distribution.

In practice, the majority of patch pairs are negative semantics. When the patch-to-patch correspondence is established, the $sr^n_{\mathcal{V}_i}$ tends to attain a large value, as the distribution $\mathcal{D}_n$ exhibits a narrow variance and a small mean, indicating a single extreme value distribution. However, if the patch t_n has multiple similar local features in volume $\mathcal{V}_i^{'}$, the Sharpe ratio may decrease. This occurs because the presence of multiple extreme values in the distribution D_n leads to an increase in both the mean and variance. Thus, the Sharpe ratio is employed to weight the patch-to-patch losses, thereby mitigating distortions in one-to-multiple relationships caused by enforcing patch-to-patch correspondence in the optimal transport process.

Method	**DICE(%)** **MSD Liver**	 **MSD Lung**	 **MSD Spleen**
S^2DC(Sinkhorn/iter10)	82.37	61.81	94.51
S^2DC(Sinkhorn/iter100)	83.29	**66.62**	95.70
S^2DC(Dual-softmax)	**83.43**	64.40	**95.73**

Table 3. Results of Sinkhorn with different iterations and dual-softmax operator. **Best** and second best are highlighted.

Algorithm 1 Differentiable Sinkhorn Algorithm in Log Domain

```
Require: Similarity matrix $\mathcal{M} \in \mathbb{R}^{n\times m}$, row and column marginals $r \in \mathbb{R}^n$, $c \in \mathbb{R}^m$, regularization parameter $\lambda > 0$, maximum iterations $T$
Ensure: Optimal transport matrix $\hat{\mathcal{M}}$
Initialize $f = \mathbf{0}_n$, $g = \mathbf{0}_m$                          ▷ Dual potentials
Set $K = \lambda \mathcal{M}$                                              ▷ Regularized cost matrix
for $t = 1$ to $T$ do
    $f \leftarrow -\lambda \log \left( \frac{r}{\exp(\frac{g}{\lambda}) \cdot \exp(\frac{K}{\lambda})} \right)$      ▷ Update row potential
    $g \leftarrow -\lambda \log \left( \frac{c}{\exp(\frac{f}{\lambda})^\top \cdot \exp(\frac{K}{\lambda})} \right)$      ▷ Update column potential
end for
Compute $\hat{\mathcal{M}} = \exp\left(\frac{f+g+K}{\lambda}\right)$      ▷ Transport matrix in log domain
return $\hat{\mathcal{M}}$
```

E. Outperform existing methods on CC-CCII

We follow previous work to pretrain on 1.6k CT data (i.e., BTCV, TCIA Covid19, and LUNA) and fine-tune on CC-CCII. The results of general SSL and medical SSL methods are shown in Tab. 4. Our methods outperform general SSL and medical SSL methods. In addition to Swin-UNETR, we also do experiments in UNETR structures. The S^2DC pre-training result in UNETR outperforms other results based on the UNETR structure.

F. Result visualization

F.1. Visualizations of downstream tasks

We visualize some segmentation results and image-to-image translation results as shown in Figure 5 and 3. From Figure 5, S^2DC shows good segmentation results on fine anatomical structures. From Figure 3, we can find T1→T2 performs worse than T1→T1ce. In the T1 sequence, water and fluids appear dark, which is the opposite of the T2 sequence. That might lead to difficulty in translation.

F.2. Visualizations of patch-to-patch correspondence.

We visualize the patch-to-patch correspondences after pre-training. As shown in Fig. 4, our method achieves more reliable alignments with fewer misalignments compared to existing approaches.

Method	Network	Accuracy(%)
From scratch		
UNETR	-	88.92
Swin-UNETR	-	88.04
	1.6k data	
With General SSL		
MAE3D	UNETR	89.47
MoCo v3	UNETR	84.95
Jiagaw	Swin-UNETR	86.88
PositionLabel	Swin-UNETR	97.54
With Medical SSL		
PCRLv1	Swin-UNETR	88.72
PCRLv2	Swin-UNETR	89.15
Rubik++	Swin-UNETR	89.23
Swin-UNETR	Swin-UNETR	89.45
SwinMM	Swin-UNETR	89.61
VoCo	Swin-UNETR	90.83
[HTML]EFEFEF S^2DC	UNETR	89.63
[HTML]EFEFEF S^2DC	Swin-UNETR	**93.85**
	10k data	
With Medical SSL		
PCRLv2	Swin-UNETR	93.07
Swin-UNETR	Swin-UNETR	94.15
SwinMM	Swin-UNETR	94.80
VoCo	Swin-UNETR	94.60
[HTML]EFEFEF S^2DC	Swin-UNETR	**95.34**

Table 4. Results of different methods on CC-CCII. Most results are drawn from [5]. We retrain methods on 10k pre-training data and fine-tune on CC-CCII. **Best** and second best are highlighted.

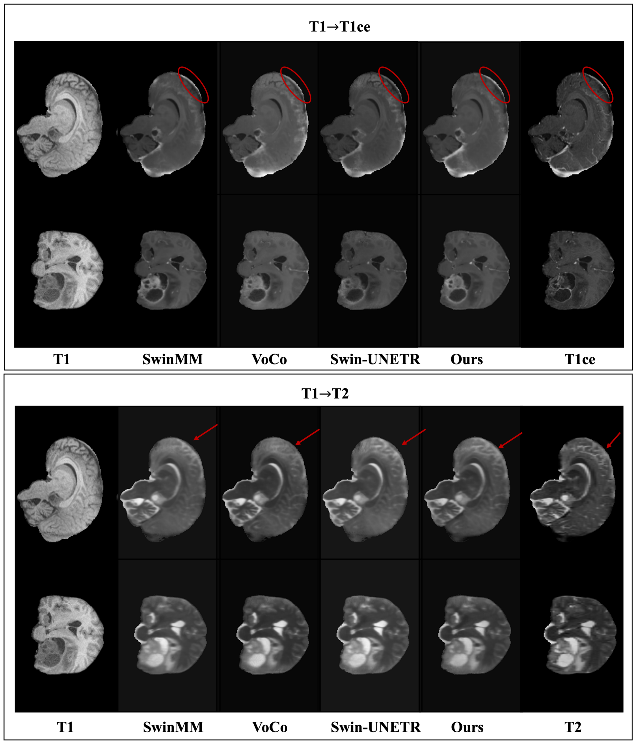

Figure 3. The visualization of image-to-image translation results of BraTs 23.

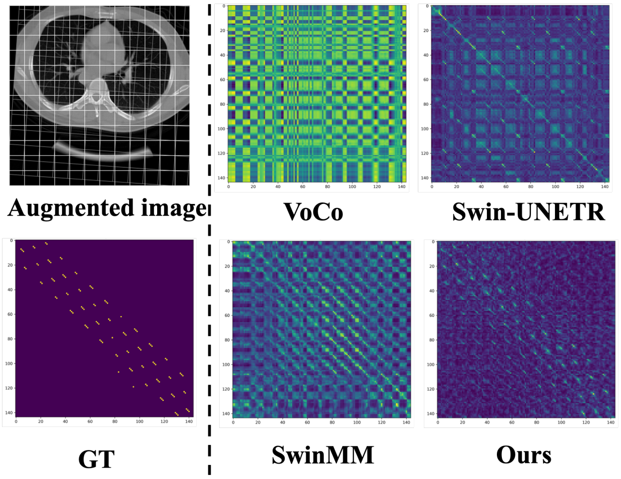

Figure 4. The visualization of patch-to-patch correspondence of different methods.

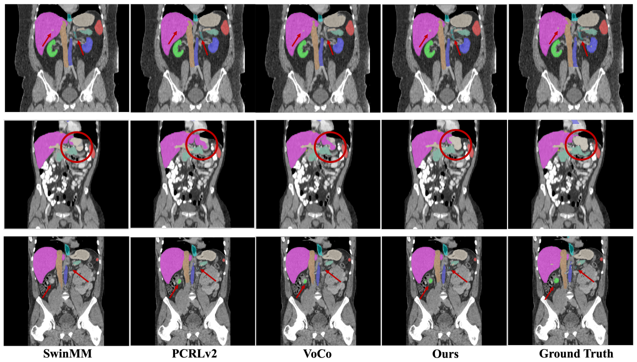

Figure 5. The visualization of segmentation results of BTCV.