Appendix

We provide the supplementary materials in the following. In Sec. A, we describe the details of datasets that are used for pre-training from public sources. In Sec. B, we illustrate the preprocessing and implementation details of fine-tuning tasks using BTCV and MSD datasets. In Sec. C, we present qualitative and quantitative comparisons of segmentation tasks in MRI modality from MSD dataset. The presented results include benchmarks from all top-ranking methods using the MSD test leaderboard. In Sec. D, the model complexity analysis is presented. Finally, we provide pseudocode of Swin UNETR self-supervised pre-training in Sec. E.

A. Pre-training Datasets

In this section, we provide additional information for our pre-training datasets. The proposed Swin UNETR is pre-trained using five collected datasets. The total data cohort contains $5,050$ CT scans of various body region of interests (ROI) such as head, neck, chest, abdomen, and pelvis. LUNA16 [47], TCIA Covid19 [18] and LiDC [2] contain 888, 761 and 475 CT scans which composes the chest CT cohort. The HNSCC [22] has $1,287$ CT scans from head and neck squamour cell carcinoma patients. The TCIA Colon dataset [29] comprises the abdomen and pelvis cohort with $1,599$ scans. We split 5% of each dataset for validation in the pre-training stage. Table S.1 summarizes sources of each collected dataset. Overall, the number of training and validation volumes are $4,761$ and 249, respectively. The Swin UNETR encoder is pre-trained using only unlabeled images, annotations were not utilized from any of theses datasets. We first clip CT image intensities from -1000 to 1000, then normalize to 0 and 1. To obtain informative patches of covering anatomies, we crop sub-volumes of $96 \times 96 \times 96$ voxels at foregrounds, and exclude full air (voxel = 0) patches. In summary, Swin UNETR is pre-trained via a diverse set of human body compositions, and learn a general-purpose representation from different institutes' data that can be leveraged for wide range of fine-tuning tasks.

B. Preprocessing Pipelines

We report fine-tuning results on two public benchmarks: BTCV [32] and MSD challenge [48]. BTCV contains 30 CT scans with 13 annotated anatomies and can be formulated as a single multi-organ segmentation task. The MSD contains 10 tasks for multiple organs, from different sources and using different modalities. Details regarding preprocessing these datasets are provided in the subsequent sub-sections of 2.1 and 2.2.

B.1. BTCV Dataset

All CT scans are interpolated into the isotropic voxel spacing of $[1.5 \times 1.5 \times 2.0]$ mm. The multi-organ segmentation problem is formulated as a 13 class segmentation, which includes large organs such as liver, spleen, kidneys and stomach; vascular tissues of esophagus, aorta, IVC, splenic and portal veins; small anatomies of gallbladder, pancreas and adrenal glands. Soft tissue window is used for clipping the CT intensities, then normalized to 0 and 1 followed by random sampling of $96 \times 96 \times 96$ voxels. Data augmentation of random flip, rotation and intensities shifting are used for training, with probabilities of 0.1, 0.1, and 0.5, respectively.

B.2. MSD Dataset

The MSD challenge contains 6 CT and 4 MRI datasets. We provide additional parameters of pre-processing and augmentation details for each task as follows:

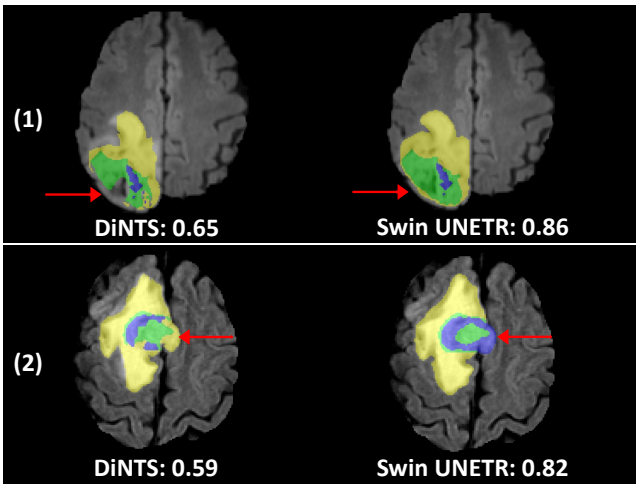
Task01 BrainTumour: The four modalities MRI images for each subject are formed into 4 channels input. We convert labels to multiple channels based on tumor classes. which label 1 is the peritumoral edema, label 2 is the GD-enhancing tumor, and label 3 is the necrotic and non-enhancing tumor core. Label 2 and 3 are merged to construct tumor core (TC), label 1, 2 and 3 are merged to construct whole tumor (WT), and label 2 is the enhancing tumor (ET). We crop the sub-volume of $128 \times 128 \times 128$ voxels and use channel-wise nonzero normalization for MRI images. Data augmentation probabilities of 0.5, 0.1 and 0.1 are set for random flips at each axis, intensities scaling and shifting, respectively.

Task02 Heart: The heart MRI images are interpolated to the isotropic voxel spacing of 1.0 mm. Channel-wise nonzero normalization is applied to each scan. We sample the training sub-volumes of $96 \times 96 \times 96$ voxels by ratio of positive and negative as 2:1. Augmentation probabilities for random flip, rotation, intensities scaling and shifting are set to 0.5, 0.1, 0.2, 0.5, respectively.

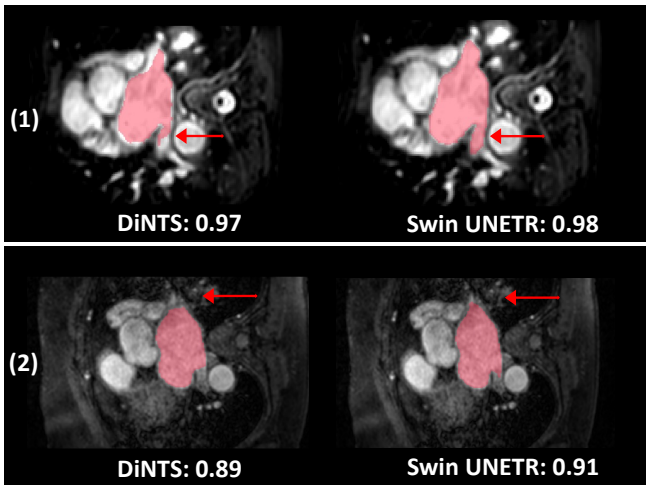
Task03 Liver: Each CT scan is interpolated to the isotropic voxel spacing of 1.0 mm. Intensities are scaled to $[-21,189]$, then normalized to $[0,1]$. 3D patches of $96 \times 96 \times 96$ voxels are obtained by sampling positive and negative ratio of $1:1$. Data augmentation of random flip, rotation, intensities scaling and shifting are used, for which the probabilities are set to 0.2, 0.2, 0.1, 0.1, respectively.

Task04 Hippocampus: Each hippocampus MRI image is interpolated by voxel spacing of $0.2 \times 0.2 \times 0.2$, then applied spatial padding to $96 \times 96 \times 96$ as the input size of Swin UNETR model. Same as other MRI datasets, channel-wise nonzero normalization is used for intensities. Probability of 0.1 is used for random flip, rotation, intensity scaling & shifting.

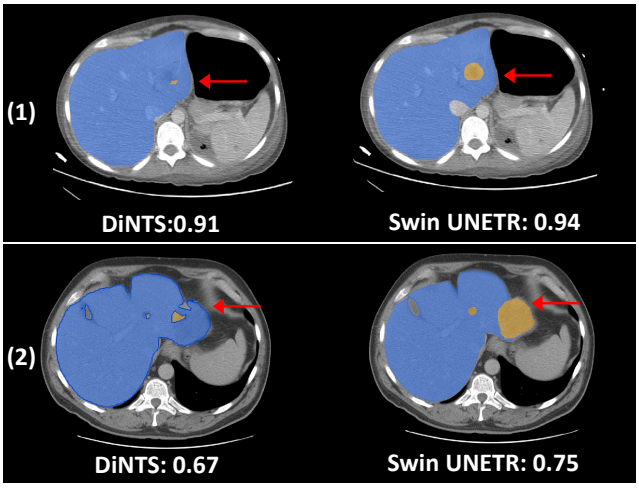
Task05 Prostate: We utilize both given modalities for prostate MRI images for each subject as two channels input. Channel-wise nonzero normalization is used. Voxel spacing of 0.5 and spatial padding of each axis are employed to construct the input size of $96 \times 96 \times 96$. We use random flip, rotation, intensity scaling and shifting with probabilities



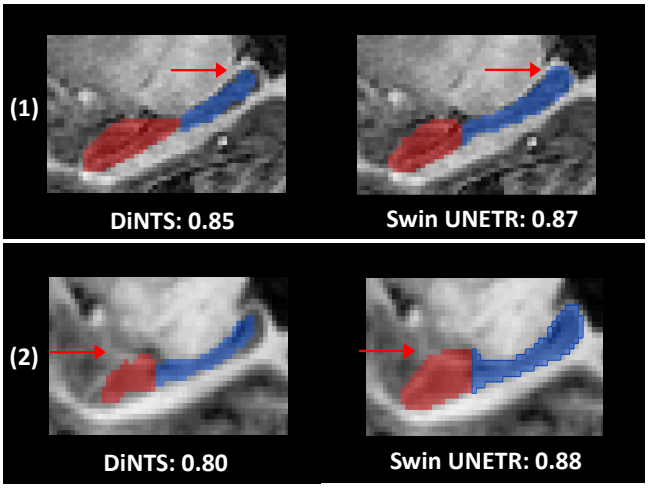
Task01 BrainTumour



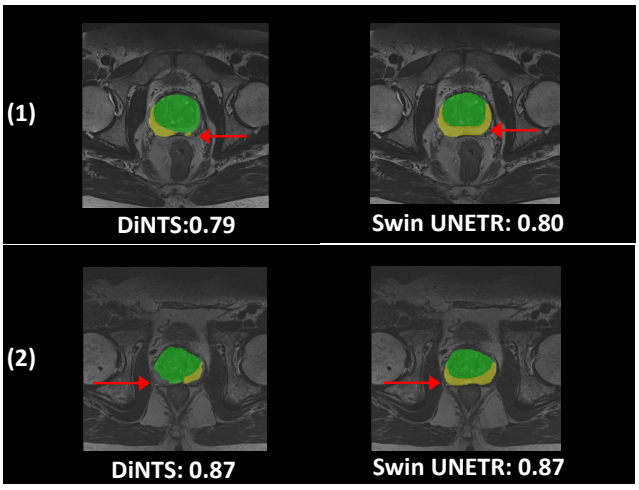
Task02 Heart



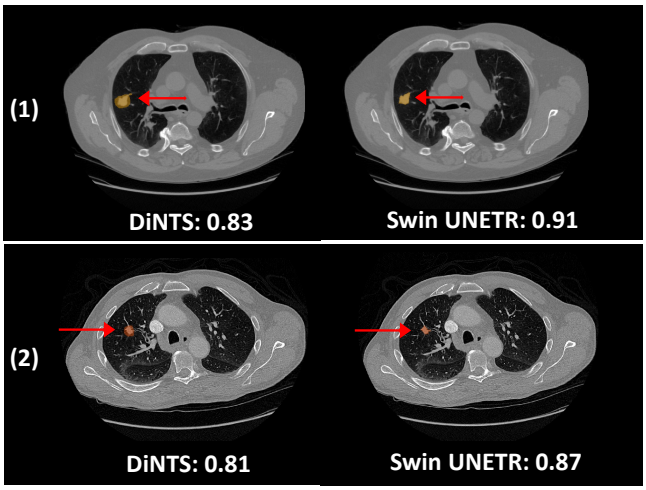
Task03 Liver



Task04 Hippocampus



Task05 Prostate



Task06 Lung

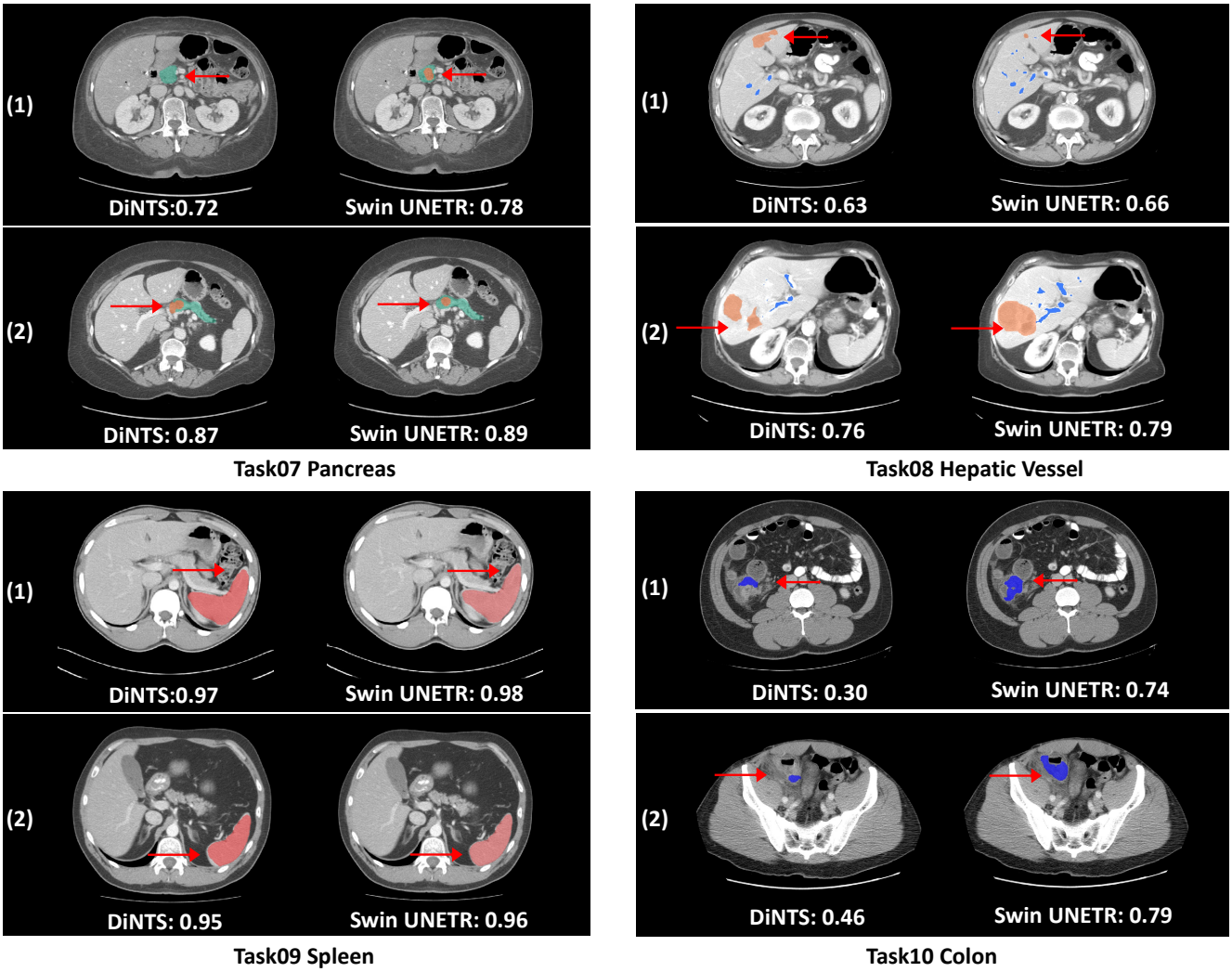


Figure S.1. Qualitative visualizations of the proposed Swin UNETR and DiNTS on MSD Tasks

Dataset	Region of Interest	#Total Samples	Source	Train/Validation
LUNA16 [47]	Chest	888	luna16.grand-challenge.org/Data/	844/44
TCIA Covid19 [18]	Chest	761	wiki.cancerimagingarchive.net/display/Public/COVID-19	723/38
HNSCC [22]	Head/Neck	1287	wiki.cancerimagingarchive.net/display/Public/HNSCC	1223/64
TCIA Colon [29]	Abdomen/pelvis	1599	www.cancerimagingarchive.net/collections/	1520/79
LiDC [2]	Chest	475	wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI	451/24

Table S.1. Summary of datasets for pre-training, the use of cohorts identifies diversified regions of interest.

of 0.5 as data augmentations. Random affine is applied as additional transformation with scale factor of $[0.3,0.3,0.0]$ and rotation range of $[0,0,pi]$ at each axis.

Task06 Lung: We interpolate each image to isotropic voxel spacing of 1.0. Houndsfield unit (HU) range of [-1000, 1000] is used and normalized to $[0,1]$. Subsequently, training sample are cropped to $96\times96\times96$ with positive and negative ratio of $2:1$. Augmentation probabilities of 0.5, 0.3, 0.1, 0.1 are used for random flip, rotation, intensities scaling and shifting.

Task07 Pancreas: We clip the intensities to a range of -87 to 199. Patch size of $96\times96\times96$ is used to sample training data with positive and negative ratio of $1:1$. We set augmentation of random flip, rotation and intensity scaling to probabilities of 0.5, 0.25 and 0.5, respectively.

Task08 HepaticVessel: To fit the optimal tissue window for hepatic vessel and tumor, we clip each CT image intensities to $[0, 230]$ HU. We apply data augmentation same with Task07 Pancreas for training.

Organ	Task02 Heart		Task04 Hippocampus						Task05 Prostate						MRI tasks Avg	
Metric	DSC1	NSD1	DSC1	DSC2	Avg.	NSD1	NSD2	Avg.	DSC1	DSC2	Avg.	NSD1	NSD2	Avg.	DSC	NSD
Kim et al [31]	93.11	96.44	90.11	88.72	89.42	97.77	**97.73**	97.75	72.64	89.02	80.83	95.05	98.03	96.54	80.96	93.43
Trans VW [23]	**93.33**	96.51	**90.29**	**88.77**	**89.53**	**97.87**	97.67	**97.77**	73.69	88.88	81.29	95.42	98.52	96.97	81.32	93.72
C2FNAS [59]	92.49	95.81	89.37	87.96	88.67	97.27	97.35	97.31	74.88	88.75	81.82	**98.79**	95.12	96.96	81.24	93.49
Models Gen [65]	**93.33**	96.51	**90.29**	**88.77**	**89.53**	**97.87**	97.67	**97.77**	73.69	88.88	81.29	95.42	98.52	96.97	81.32	93.72
nnUNet [28]	93.30	**96.74**	90.23	88.69	89.46	97.79	97.53	97.75	**76.59**	**89.62**	**83.11**	96.27	**98.85**	**97.56**	81.74	93.91
DiNTS [27]	92.99	96.35	89.91	88.41	89.16	97.76	97.56	97.66	75.37	89.25	82.31	95.96	98.82	97.39	81.76	94.03
SwinUNETR	92.62	96.23	89.95	88.42	89.19	97.63	97.32	97.48	75.65	89.15	82.40	95.89	98.70	97.30	**82.14**	**94.66**

Table S.2. Additional MSD MRI test dataset performance comparison of Dice and NSD. Benchmarks obtained from MSD test leaderboard. Task01 BrainTumuor results are shown in the paper. Note: The results reported for TransVW [23] and Models Genesis [65] are from the official leaderboard for MRI tasks.

Models	#Params (M)	FLOPs (G)	Inference Time (s)
nnUNet [28]	19.07	412.65	10.28
CoTr [55]	46.51	399.21	19.21
TransUNet [7]	96.07	48.34	26.97
ASPP [10]	47.92	44.87	25.47
SETR [61]	86.03	43.49	24.86
UNETR	92.58	41.19	12.08
SwinUNETR	61.98	394.84	13.84

Table S.3. Comparison of number of parameters, FLOPs and averaged inference time for various models in BTCV experiments.

Task09 Spleen: Spleen CT scans are pre-process with interpolation isotropic voxel spacing of $1.0\ mm$ on each axis. Soft tissue window of $[-125,275]$ HU is used for the portal venous phase contrast enhanced CT images. We use the training data augmentation of random flip, intensity scaling & shifting with probabilities of 0.15, 0.1, and 0.1, respectively. **Task10 Colon:** We use HU range of $[-57,175]$ for the colon tumor segmentation task and normalized to 0 and 1. Next, we sample training sub-volumes by positive and negative ratio of $1:1$. Same as Task07 and Task08, we use random flip, rotation, intensity scaling as augmentation transforms with probabilities of 0.5, 0.25 and 0.5, respectively.

C. Results

C.1. MSD Qualitative Comparisons

In this section, we provide extensive segmentation visualization from MSD dataset. In particular, we compare two cases randomly selected from Swin UNETR and DiNTS for each MSD task. As shown in Fig S.1, DiNTS includes the under-segmentation due to lack of parts of labels (Heart, Hippocampus). The missing parts result in a lower Dice score. On BrainTumour, Liver, Pancreas, HepaticVessel and Colon tasks, the comparison indicate that our method achieves better segmentation where the under-segmentation of tumors are observed in DiNTS. For Lung task, the over-segmentation is observed with DiNTS where surrounding tissues are included with label of the lung cancer, while Swin UNETR clearly delineate the boundary. In Heart and Spleen, DiNTS and Swin UNETR have comparable Dice score, yet Swin UNETR performs better segmentation on tissue corner (See Fig S.1). Overall, Swin UNETR achieves better segmentation results and solves the under- and over-segmentation outliers as observed in segmentation via DiNTS.

C.2. MSD Quantitative Comparisons

In this section, we provide the quantitative benchmarks of MRI segmentation tasks from MSD dataset. In addition to Task01 BrainTumour, we implement experiment on three remaining MRI dataset including Heart, Hippocampus and Prostate (see Table. S.2). The results are directly obtained from the MSD[6] leaderboard. Regarding MRI benchmark, we achieve much better performance on brain tumor segmentation presented in the paper, with average Dice improvement of 2% against second best performance. Comparing to models genesis [65], nnUNet [28], the Swin UNETR shows comparable results on Heart, Hippocampus and Prostate. Overall, we achieve the best average results (Dice of 82.14% and NSD of 94.66%) across four MRI datasets, showing Swin UNETR's superiority of medical image segmentation.

D. Model Complexity and Pre-training Time

In this section, we examine the model complexity along with inference time. In Table. S.3, the number of network paramerts, FLOPs, and averaged inference time of Swin UNETR and baselines on BTCV dataset are presented. We calculate the FLOPs and inference time based on input size of $96 \times 96 \times 96$ used in the BTCV experiments with sliding window approach. Swin UNETR shows moderate size of parameter with 61.98M, less than transformer-based methods such as TransUNet [7] of 96.07M, SETR [62] of 86.03M, and UNETR [24] of 92.58M, but larger than 3DUNet (nnUNet) [28] of 19.07M, ASPP [10] 47.92M. Our model also shows comparable FLOPs and inference time in terms of 3D approaches such as nnUNet [28] and CoTr [55]. Overall, Swin UNETR outperforms CNN-based and other transformer-based methods while perserves moderate model complexity.

[6] https://decathlon-10.grand-challenge.org/evaluation/challenge/leaderboard/

Algorithm S.1 Pytorch Pseudocode of Swin UNETR Self-Supervised Pre-training.

```
# RandRot:  transforms of random rotation
# Cutout:  transforms of cutout
# Encoder:  swin transformer encoder
# RecHead:  reconstruction head
# RotHead:  rotation head
# CnHead:  contrastive head
# Linpaint:  reconstruction loss
# Lrot:  rotation loss
# Lcontrast:  contrastive loss
for x in Loader:  # minibatch of samples
   x1, rot1 = RandRot(x)
   x2, rot2 = RandRot(x)
   x1', x2' = Cutout(x1), Cutout(x2)
   z1, z2 = Encoder(x1'), Encoder(x2')
   rec1, rec2 = RecHead(z1), RecHead(z2)
   contr1, contr2 = CnHead(z1), CnHead(z2)
   r1, r2 = RotHead(z1), RotHead(z2)
   rot, r = torch.cat(rot1, rot2), torch.cat(r1, r2)
   rec, x = torch.cat(rec1, rec2), torch.cat(x1, x2)
   loss = Linpaint(rec,x) + Lrot(r,rots) + Lcontrast(contr1,contr2)
   loss.backward() # back-propagate
```

Regarding self-supervised pre-training time of Swin UNETR encoder, our approach takes only approximately 6 GPU days. We evaluate pre-training on the 5 collected public datasets with totally 5,050 scans for training and validation, and set maximum training iterations to 45K steps.

E. Pre-Training Algorithm Details

In this section, we illustrate the Swin UNETR pre-training details. The Pytorch-like pseudo-code implementation is shown in Algorithm S.1. The Swin UNETR is trained in self-supervised learning paradigm, where we design masked volume inpainting, rotation prediction and contrastive coding as proxy tasks. The self-training aims at improving the quality of representations learnt by large unlabeled data and propagating to smaller fine-tuning dataset. To this end, we leverage multiple transformations for input 3D data, which can exploit inherent context by a mechanism akin to autoencoding and similarity identification. In particular, given an input mini batch data, the transform of random rotation is implemented on each image in the mini batch iteratively. To simultaneously utilize augmentation transformations for contrastive learning, the random rotation of $0^\circ, 90^\circ, 180^\circ, 270^\circ$ is applied twice on the same input to generate randomly augmented image pairs of the same image patch. Subsequently, the mini batch data pairs are constructed with the cutout transforms. The drop size of voxels are set to 30% of input sub-volumes. We randomly generate masked ROIs inside image, until the total masked voxels are larger than scheduled number of dropping voxels. Unlike canonical pre-training rules of masked tokens in BERT [19], our local transformations to the CT sub-volumes are then arranged to neighbouring tokens. This scheme can construct semantic targets across partitioned tokens, which is critical in medical spatial context. By analogy to Models Genesis [65], which is CNN-based model consisting expensive convolutional, transposed convolution layers and skip connection between encoder and decoder, our pre-training approach is trained to reconstruct input sub-volumes from the output tokens of the Swin Transformer. Overall, the intuition of modeling inpainting, rotation prediction and contrastive coding is to generalize better representations from aspects of images context, geometry and similarity, respectively.