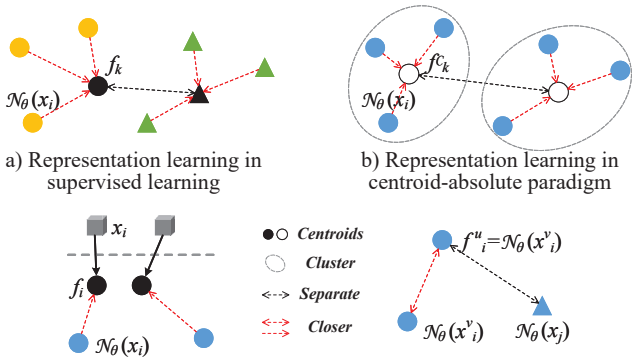
Geometric Visual Similarity Learning in 3D Medical Image Self-Supervised Pre-training (Supplementary Material)

Yuting He[1], Guanyu Yang[1]*, Rongjun Ge[2], Yang Chen[1], Jean-Louis Coatrieux[3], Boyu Wang[4], Shuo Li[5]
[1]Southeast University [2]Nanjing University of Aeronautics and Astronautics
[3]University of Rennes 1 [4]Western University [5]Case Western Reserve University

A Rethink GVSL and representation learning

Our GVSL is an unsupervised representation learning paradigm which constructs a geometric metric to learn the inter-image similarity, thus achieving a consistent representation for same semantic regions based on a reliable semantics' correspondence.



a) Representation learning in supervised learning

b) Representation learning in centroid-absolute paradigm

c) Representation learning in centroid-hidden paradigm

d) Representation learning in centroid-relative paradigm

Figure 1. The view of the embedding space. a) The representation learning in supervised learning gather features to the centroids $\mathbf{f}_{1:K}$ corresponding to their classes, and separate the centroids. b) Centroid-absolute paradigm clusters features for centroids $\mathbf{f}^c_{1:K}$, and learns to gather features to these centroids. c) Centroid-hidden paradigm generates the pretext labels via manual designed methods and learns follow the supervised learning. d) Centroid-relative paradigm train to gather the features of same image's different views, and separate the features of different images.

A.1 Representation in supervised learning

Let's start by rethinking supervised learning from labeled dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^I, y_i \in \mathbf{y}_{1:K}$, where x_i and y_i are the i_{th} image and label, and I is the number of data, K is the number of classes. The whole framework can be divide into two parts, including the learning of representation $\mathcal{N}_\theta$ with parameters θ and the learning of specific task $\mathcal{G}_\xi$ with parameters ξ [8]. The representation part $\mathcal{N}_\theta$ maps images to an embedding space for features, and the specific task part $\mathcal{G}_\xi$ maps the features in the embedding space to the task space. The supervised learning train the network to learning the representation and specific task via minimizing the distance d of framework's outputs and labels following

$$\min_{\theta,\xi} d(\mathcal{G}_\xi(\mathcal{N}_\theta(x_i)), y_i)). \tag{1}$$

We assume that there is a centroid $f_k \in \mathbf{f}_{1:K}$ in the embedding space that makes $\mathcal{G}_\xi(f_k) = y_i$, then the Equ.1 is equivalent to

$$\begin{aligned} &\min_{\theta,\xi} d(\mathcal{N}_\theta(x_i), f_k) \\ &s.t. \quad \mathcal{G}_\xi(f_k) = y_i. \end{aligned} \tag{2}$$

Obviously, in this process (Fig.1 a)), the representation part $\mathcal{N}_\theta$ is trained to gather features to the centroids $\mathbf{f}_{1:K}$ corresponding to their classes via the specific task part $\mathcal{G}_\xi$. Therefore, the learning of $\mathcal{G}_\xi$ optimizes the centroids $\mathbf{f}_{1:K}$ in the embedding space to distinguish different classes, and the learning of $\mathcal{N}_\theta$ optimizes the clustering effect of same class data.

A.2 Learning representation without annotation

When labels are unavailable $\mathcal{D} = \{x_i\}_{i=1}^I$, this means the centroids $\mathbf{f}$ in the embedding space are unavailable to guide the clustering effect. Therefore, the self-supervised representation learning [18] targets building the centroids $\mathbf{f}$ via pretext tasks, thus guiding the network learning potential clustering effect. According to the difference of $\mathbf{f}$, the existing methods can be divided into three paradigms:

- Centroid-hidden paradigm (Fig.1 c)) [16, 18]: This paradigm still follows the Equ.2, and generates the pretext labels via designed transformation methods $\mathcal{T}$ (e.g., restoration [22], rotation [16]). Therefore, like the supervised learning, this paradigm impliedly creates centroids $\mathbf{f}$ in the embedding space according to the pretext labels, learns the $\mathcal{N}_\theta$ to gather features to

*Corresponding author: yang.list@seu.edu.cn

the centroids and learns the $\mathcal{G}_\xi$ to distinguish the centroids $\mathbf{f}$ in embedding space.

$$\begin{aligned} &\min_{\theta,\xi} d(\mathcal{N}_\theta(x_i), f_i) \\ &s.t. \quad \mathcal{G}_\xi(f_i) = \mathcal{T}(x_i). \end{aligned} \tag{3}$$

* *Observation*: The centroids extremely depend on manual defined transformation methods $\mathcal{T}$, which will bring large bias in the representation. For example, the rotation method [16] will make the $\mathcal{N}_\theta$ biased to the position features, and some images whose positions are semantics-independent information will be mis-represented.

- Centroid-absolute paradigm (Fig.1 b)) [2, 17]: This paradigm utilizes the clustering methods $\mathcal{C}^K$ (K is the number of clustered centroids) to discover the clustering patterns of features, thus building the centroids $\mathbf{f}^{\mathcal{C}}_{1:K}$ and gathering the represented features to these centroids, like DeepCluster [2]

$$\begin{aligned} &\min_{\theta} d(\mathcal{N}_\theta(x_i), f^{\mathcal{C}}_k) \\ &s.t. \quad f^{\mathcal{C}}_k = \mathcal{C}^K(\mathcal{N}_\theta(x_i); \mathcal{D}). \end{aligned} \tag{4}$$

* *Observation*: The clustering method $\mathcal{C}^K$ is the bottleneck in this paradigm. The existing works [2, 17] utilize K-means [9] as the clustering methods which is extremely interfered by images' semantic-independent variations. Therefore, the clustered centroids will bring imprecise information, finally learning mis-representation.

- Centroid-relative paradigm (Fig.1 d)) [3, 4]: This paradigm has no explicit centroids $\mathbf{f}$, but train $\mathcal{N}_\theta$ to contrast images. A popular method is contrastive learning [3, 4, 10]. This method constrains the representation of same image's different views (x^v_i, x^u_i) to be consistent and different images (x_i, x_j) to be separated, thus gaining clustering effect under the training of big data.

$$\begin{aligned} &\min_{\theta} d(\mathcal{N}_\theta(x^v_i), f^u_i) - d(\mathcal{N}_\theta(x_j), f^u_i) \\ &s.t. \quad f^u_i \triangleq \mathcal{N}_\theta(x^u_i) \end{aligned} \tag{5}$$

* *Observation*: This paradigm have to learn inner-image similarity and inter-image dissimilarity. However, if the images share numerous same semantics, this paradigm will make the $\mathcal{N}_\theta$ learn the task-unconcerned features. Especially in our task, the 3D medical images share numerous same semantic regions due to the consistency of human anatomies, the direct learning of separation will mislead the consistent representation of these same semantic regions. Although some works have removed the learning of inter-image dissimilarity, the single learning of inner-image similarity will bring the risk of dimensional collapse [13].

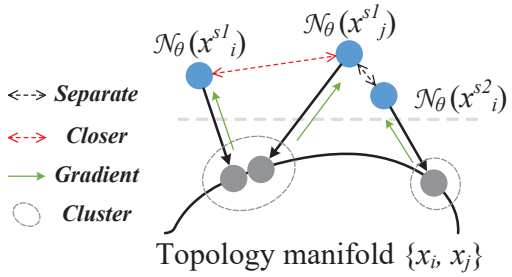


Figure 2. GVSL in the view of embedding space. It projects features onto a manifold of consistent topology, and gathers the semantic features ($\mathcal{N}_\theta(x^{s1}_i), \mathcal{N}_\theta(x^{s1}_j)$, $s1$ means the semantic regions on images) which are closed on this manifold.

Conclusion: Observing these three paradigms, we can draw three conclusions:

- *A self-discovery method* to drive the learning of clustering effect is crucial to avoid the large bias caused by the manual designed transformation.
- *Prior knowledge of semantics* is crucial to avoid the interference caused by images' semantic-independent variations during the self-discovery of clustering effect.
- *Learning inter-image similarity* is crucial for 3D medical image self-supervised pre-training.

Therefore, our GVSL fuses the prior of topological invariance into the learning of inter-image similarity in a self-discovery process, achieving power self-supervised pertraining.

A.3 Learning GVSL

Our GVSL embeds a geometric mapping into the measurement of different images, bringing three advancement compared with above three paradigms:

- Compared with the centroid-hidden paradigm, it brings a self-discovery process which learns a geometric matching head $\mathcal{G}_\xi$ to discover the corresponding of visual objects between images to learn consistent representation of same semantics.
- Compared with the centroid-absolute paradigm, it embeds the prior of topological invariance into the discovery of correspondence, avoiding the interference caused by images' semantic-independent variations.

- Compared with the centroid-relative paradigm, it avoids the direct learning of inter-image dissimilarity in global, and utilizes the geometric matching to discover the correspondence of same semantic regions inner two images and learn consistent representation of them.

Compared with the Equ.5, GVSL (Equ.6) takes a $\mathcal{G}_\xi$ to discover the correspondence of same semantic regions between two images, avoiding the direct enlarging of feature distance for two images in Equ.5.

$$\begin{aligned} &\min_{\theta,\xi} d(\mathcal{G}_\xi(\mathcal{N}_\theta(x_i), f_j; \{x_i, x_j\})) \\ &s.t. \quad f_j \triangleq \mathcal{N}_\theta(x_j) \end{aligned} \tag{6}$$

For the $\mathcal{G}_\xi$, it is a learnable metric which is embedded the prior of topological invariance. It embeds the two original 3D medical images x_i, x_j which have consistent topology (Introduction section) into the calculation of the distance, and models the measurement of the distance for two features $f_i \triangleq \mathcal{N}_\theta(x_i), f_j$ as the measurement of the alignment degree for two image x_i, x_j. Therefore, as shown in FIg.2, this implicitly projects features onto a manifold of consistent topology (the invariant distribution of semantic regions in 3D medical images $\{x_i, x_j\}$), and gathers the semantic features $(\mathcal{N}_\theta(x_i^{s1}), \mathcal{N}_\theta(x_j^{s1}), s1$ means the semantic regions on images) which are closed on this manifold.

B Algorithm

As illustrated in Alg.1, our GVSL framework learns the GM between two images and the self-restoration as a baseline for consistent representation of same semantics between images.

C Details of Transformation Operation $\mathcal{T}$

During self-supervised training, our GVSL uses the consistency of following image transformation operations:

- Random in-painting: This operation randomly selects 3D boxes inner images and the contents of these regions are replaced by the noise from a uniform distribution. Therefore, when learning the self-restoration and our GVSL, the network $\mathcal{N}_\theta$ will learn the dependency between the semantics and their context.

- Random local-shuffling: This operation randomly selects 3D boxes inner images and shuffles the voxels in the box regions. Therefore, when learning the self-restoration and our GVSL, the network $\mathcal{N}_\theta$ will learn the representation of texture features for semantics.

- Random non-linear transformation: This operation uses Bézier Curve which assigns every voxel a unique value via transform the distribution function of image. Therefore, the network $\mathcal{N}_\theta$ will learn the intensity information of semantic regions during the learning of self-restoration and our GVSL.

More specific related introductions can be find in the paper [22] which we follows. The Fig.3 demonstrates the transformation operations visually.

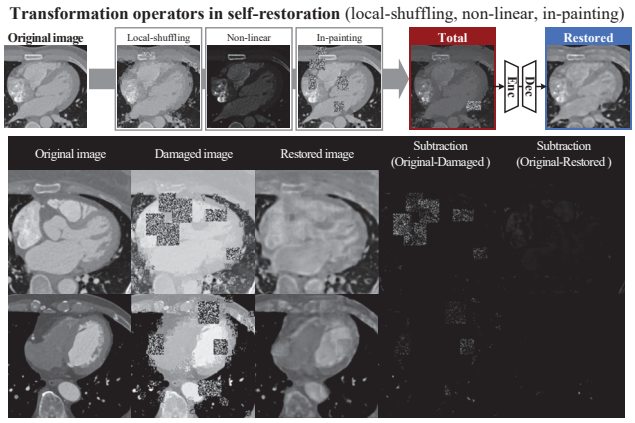


Figure 3. The visualization of the transformation operations. We utilize the in-painting, local-shuffling, and non-linear to construct the transformation distribution $\mathcal{T}$.

D Details in our GVSL

D.1 Details in spatial transformation

We utilize the spatial transformation following [12] which is the function *torch.nn.functional.grid_sample* in PyTorch. For each voxel p in image x, the DVF ψ displaces the p to a new (subpixel) voxel location $\psi(x(p))$ in image space. Then, the voxel in subpixel position is linearly interpolated to a near integer location at eight neighboring voxels. This process is formulated as

$$\psi(x(p)) = \Sigma_{q \in \psi(\mathbb{Z}(p))} x(q) \Pi_{d \in \{x,y,z\}} (1 - |\psi_d(x(p)) - q_d|), \tag{7}$$

where $\psi(\mathbb{Z}(p))$ are the voxel neighbors of $\psi(x(p))$, $\{x, y, z\}$ are the x, y, z axes of 3D image.

D.2 Details in the network $\mathcal{N}_\theta$

We utilize the 3D U-Net [5] which is widely used in 3D medical images as the backbone network $\mathcal{N}_\theta$ in our framework. Owing to the limitation of GPU memory, we only use the batch size of 1 in our transferring process, and the batch size of 2 in our pre-training process. To avoid the overfitting problem caused by the Batch Normalization (BN) [11], we utilize the Group Normalization [21] to replace the BN in the original network.

Algorithm 1: GVSL: **G**eometric **V**isual **S**imilarity **L**earning

Input:

$\mathcal{D}, \mathcal{T}$ dataset and the distribution of transformations;
$\theta, \mathcal{N}_\theta$ initial parameters for backbone network, backbone network;
$\xi, \mathcal{G}_\xi$ initial parameters for GM, GM head;
$\iota, \mathcal{R}_\iota$ initial parameters for self-restoration, restoration head;
$optimizer$ optimizer, updates parameters via gradient;
K, N, η iteration number, batch size, and learning rate.

for $k = 1...K$ **do**
  $\mathcal{B} \leftarrow \{\{x_A^i, x_B^i\} \sim \mathcal{D}\}_{i=1}^N$; // sample two batches from dataset
  for $i, \{x_A, x_B\} \in \mathcal{B}$ **do**
    $t \sim \mathcal{T}$; // sample image transformation
    $x_A^t \leftarrow t(x_A)$; // transform image x_A
    $\{f_A^g, f_A^l\} \leftarrow N_\theta(x_A^t)$ and $\{f_B^g, f_B^l\} \leftarrow N_\theta(x_B)$; // compute global and local features from two images
    $x_A^{'} \leftarrow \mathcal{R}_\iota(f_A^l)$; // restore image x_A in restoration head
    $\psi_{AB} \leftarrow \mathcal{G}_\xi(f_A^l, f_B^g, f_A^g, f_B^g)$; // estimate a displacement vector field
    $x_{AB} \leftarrow \psi_{AB}(x_A)$; // align image x_A to x_B
    $l_{\theta,\xi}^{GVSL,i} \leftarrow l_{\theta,\xi}^{NCC}(x_{AB}, x_B) + l_{\theta,\xi}^{smooth}(\psi_{AB})$; // calculate the NCC loss and smooth loss for GVSL
    $l_{\theta,\iota}^{MSE,i} \leftarrow \|x_A^{'} - x_A\|^2$; // calculate the MSE loss for self-restoration
  end
  $\delta\theta \leftarrow \frac{1}{N}\sum_{i=1}^N (\partial_\theta l_{\theta,\xi}^{GVSL,i} + l_{\theta,\iota}^{MSE,i})$;
  $\delta\xi \leftarrow \frac{1}{N}\sum_{i=1}^N \partial_\xi l_{\theta,\xi}^{GVSL,i}$;
  $\delta\iota \leftarrow \frac{1}{N}\sum_{i=1}^N \partial_\iota l_{\theta,\iota}^{MSE,i}$; // compute the loss gradient w.r.t. θ, ξ, and ι
  $\theta \leftarrow optimizer(\theta, \delta\theta, \eta)$;
  $\xi \leftarrow optimizer(\xi, \delta\xi, \eta)$;
  $\iota \leftarrow optimizer(\iota, \delta\iota, \eta)$; // update parameters i.e. θ, ξ, and ι
end
Output: $\mathcal{N}_\theta$; // the pre-trained backbone networks

D.3 Details in the fusion operation for DVF $\odot$

As demonstrated in Equ.8, the affine matrix [1] utilizes the matrix consists of the rotation matrix, scaling matrix, shearing matrix, and translation matrix to make a movement for each voxels, thus achieving a global spatial transformation. This affine matrix ψ_{AB}^g multiplies the position index $p = \{p_x, p_y, p_z\}$ of the voxel in image grid for the affine transformed position index $\hat{p} = \{p_x, p_y, p_z\}$. The transformed position index $\hat{p}$ is subtracted to the original position index p for the affine vector and the affine vector is further added to the deformation vector in the position index $\hat{p}$ of deformation field ψ_{AB}^l (Equ.9), thus achieving the vector to move the voxel in position $\psi_{AB}(p)$. This operation is performed for whole positions in the image grid, fusing the affine matrix and the deformation field for the DVF ψ_{AB}.

E Details of Datasets and Implementations in Experiment

As shown in Tab.1, we pre-train the network on the pre-training dataset and evaluate the models on four downstream tasks with different properties, giving a complete evaluation.

E.1 Details of the pre-training dataset

The pre-training dataset consists of 302 cardiac CT images with numerous semantic regions. These images were scanned on a SOMATOM Definition Flash and the contrast media was injected during the scanning process. The x/y-resolution of these CT images is between 0.25 to 0.57 mm/voxel and the slice thickness is between 0.75 to 3 mm/voxel. The x/y-size of the images is 512 voxels and the z-size is between 128 to 994 voxels. For pre-processing, we firstly resample the resolution of these images to $1mm \times$

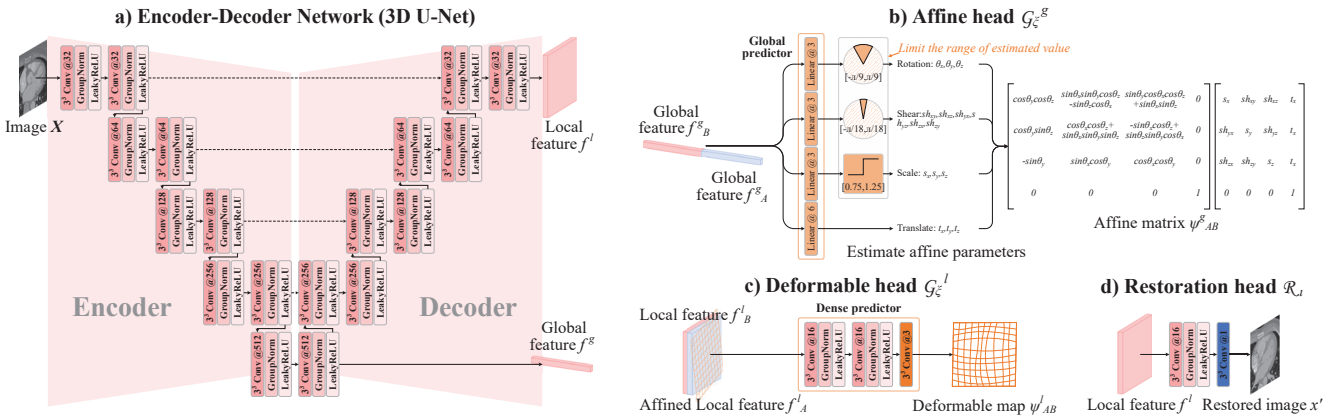


Figure 4. **The details in our framework.** a) We take the 3D U-Net as our backbone, the features from the bottleneck and the final layer are the global features f^g and the local features f^l. b) Affine head utilizes four linear layers to estimate affine parameters of rotation, shear, scale and translation. These parameters are used to make an affine matrix $\psi^{'}$ for affine transformation. c) Deformable head takes two Conv-groups followed by a convolution to estimate the deformable map $\phi^{'}$ via the local features. d) The restoration head takes a Conv-group followed by a convolution to restore the image.

$$\psi^g_{AB} = \overbrace{\begin{bmatrix} \cos\theta_y\cos\theta_z & \sin\theta_x\sin\theta_y\cos\theta_z - \sin\theta_z\cos\theta_x & \sin\theta_y\cos\theta_x\cos\theta_z + \sin\theta_x\cos\theta_z & 0 \\ \cos\theta_y\sin\theta_z & \cos\theta_x\cos\theta_z + \sin\theta_x\sin\theta_y\sin\theta_z & -\sin\theta_x\cos\theta_z + \sin\theta_z\sin\theta_y\cos\theta_x & 0 \\ -\sin\theta_y & \sin\theta_x\cos\theta_y & \cos\theta_x\cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{\text{Rotation}} \underbrace{\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Scaling}} \underbrace{\begin{bmatrix} 1 & sh_{yx} & sh_{zx} & 0 \\ sh_{xy} & 1 & sh_{zy} & 0 \\ sh_{xz} & sh_{yz} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Shearing}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Translation}}, \tag{8}$$

$$\psi_{AB}(p) \triangleq \psi^l_{AB}(p) \odot \psi^g_{AB} = \begin{bmatrix} \psi_{AB}(p_x) \\ \psi_{AB}(p_y) \\ \psi_{AB}(p_z) \end{bmatrix} = \begin{bmatrix} \psi^l_{AB}(\hat{p_x}) \\ \psi^l_{AB}(\hat{p_y}) \\ \psi^l_{AB}(\hat{p_z}) \end{bmatrix} + \begin{bmatrix} \hat{p_x} \\ \hat{p_y} \\ \hat{p_z} \end{bmatrix} - \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \begin{bmatrix} \hat{p_x} \\ \hat{p_y} \\ \hat{p_z} \\ 1 \end{bmatrix} = \psi^g_{AB} \times \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \tag{9}$$

$1mm \times 1mm$ for a unified resolution, then threshold their grayscale value to [0, 2048] and normalize them to [0, 1] for unified intensity.

E.2 Details of the downstream datasets

E.2.1 The SHC task targets on segmenting seven large heart structures on the CT images from MM-WHS 2017 dataset [23] which originally has 20 image-label pairs and 40 unlabeled images. We randomly split 15 of the image-label pairs as the training set, 5 of them as the validation set, and the original 40 unlabeled images as the testing set and test the results on the officially provided software. For pre-processing, we firstly crop the heart regions of interests (ROIs) to reduce the size due to the limited GPU memory and resample the resolution of these images to $1mm \times 1mm \times 1mm$ for a unified resolution. These images are further thresholded to [0, 2048] grayscale value, and normalized to [0, 1] via dividing by 2048 for unified intensity. This task evaluate the **inner**-scene transferring ability of the models on a **dense** prediction task for **large** structures.

E.2.2 The SAC task targets on segmenting the small coronary arteries on the Coronary CT Angiography (CCTA) images from ASOCA dataset [7] which originally has 40 image-label pairs. We randomly split 15 of them as the training set, 5 of them as the validation set, and 20 of them as the testing set. Following the SHC task, we also crop the heart ROIs, resample their resolution to $1mm \times 1mm \times 1mm$, threshold the grayscale to [0, 2048] and normalize

Table 1. The details of the clinical dataset in our pretext task and four public available datasets in our downstream tasks.

a) The details of four public available datasets in **downstream tasks**				
Name	**Target dataset**	**Train/Val/Test**	**Downstream task**	**Pre-processing**
SHC	MM-WHS 2017 CT[a]	15/5/40	Segmentation of 7 heart structures	1.Crop the heart regions 2.Resample the resolution to $1mm^3$ 3.Normailze via $\frac{\max(\min(0,x),2048)}{2048}$
SAC	ASOCA 2020 CT[b]	15/5/20	Segmentation of coronary artery	1.Crop heart regions 2.Resample the resolution to $1mm^3$ 3.Normailze via $\frac{\max(\min(0,x),2048)}{2048}$
SBM	CANDI MR[c]	40/20/43	Segmentation of 28 brain tissues	1.Crop $160^2 \times 128$ regions around brain 2.Resample the resolution to $1mm^3$ 3.Normailze via $\frac{x-\min(x)}{\max(x)-\min(x)}$
CCC	STOIC CT[d]	1000/400/600	Diagnosis of COVID-19	1.Extract lung regions via lungmask[e] 2.Resample the resolution to $1mm^3$ 3.Normailze via $\frac{\max(\min(0,x),2048)}{2048}$
b) The details of the clinical dataset in **pretext task**				
Amount	**Image type**	**Detail information**		**Pre-processing**
302	Coronary CT angiography	1.Scanner: SOMATOM Definition Flash 2.x/y-resolution: 0.25∼0.57 mm/voxel 3.Slice thickness: 0.75∼3 mm/voxel 4.x/y-size: 512 voxels, z-size: 128∼994 voxels		1.Resample the resolution to $1mm^3$ 2.Normailze via $\frac{\max(\min(0,x),2048)}{2048}$

[a] MM-WHS 2017: http://www.sdspeople.fudan.edu.cn/zhuangxiahai/0/mmwhs/
[b] ASOCA: https://asoca.grand-challenge.org/
[c] CANDI: https://www.nitrc.org/projects/candi_share/
[d] STOIC challenge: https://stoic2021.grand-challenge.org/stoic-db/
[e] Lungmask code: https://github.com/JoHof/lungmask

the intensity to [0, 1] via dividing by 2048. This task evaluate the **inner**-scene transferring ability of the models on a **dense** prediction task for **small** structures.

E.2.3 The SBM task targets on segmenting 28 brain tissues on the brain T1-weighted MR images from CANDI dataset [14] which has 103 image-label pairs. We randomly split 40 of them as the training set, 20 of them as the validation set, and 43 of them as the testing set. Following some works [6, 20] on this dataset, we crop a $160 \times 160 \times 128$ region around the center of the brain which contain the whole brain for computation efficiency. The grayscale value of these images are further limited bigger than 0, and normalized to [0, 1] via min-max normalization for unified intensity. This task evaluate the **inter**-scene transferring ability of the models on a **dense** prediction task for **multiple** (28) structures.

E.2.4 The CCC task targets on classifying (diagnosis) the COVID-19 or the health on chest CT images from the STOIC challenge dataset [19] which originally has 2000 public training set. To evaluate the models in our experiment, we further randomly split 1000 of them as the training set, 400 of them as the validation set, and 600 of them as the testing set. For pre-processing, we extract the lung regions via the existing open released code of lungmask to remove the interruption of the background, crop the lung ROIs to reduce the size, and resample the resolution of these images to $1mm \times 1mm \times 1mm$ for a unified resolution. Following the SHC task, we also threshold the grayscale to [0, 2048] and normalize the intensity to [0, 1] via dividing by 2048. This task evaluate the **inner**-scene transferring ability of the models on a **global** prediction task.

E.3 Implementation details of transfer learning on downstream tasks

E.3.1 Implementation for linear evaluation We take linear evaluation to evaluate the clustering effect of the extracted features thus demonstrating the representability of the pre-trained network. 1) For segmentation tasks (SHC, SAC, SBM), we use the whole pre-trained backbone network $\mathcal{N}_\theta$ as a fixed feature extractor for the new downstream datasets. And then, the local features f^l from the decoder of the network are used to train a convolutional layer followed with a Softmax activation function. 2) Like the implementation for segmentation tasks, for classification task (CCC), we use the encoder part of the backbone network $\mathcal{N}_\theta$ as a fixed feature extractor for downstream tasks. And then, the global features f^g from the fixed encoder are used to train a linear layer followed with a Sigmoid activation function in CCC task for the evaluation of the representability for global features. We use a batch size of 1 due to the limitation of GPU memory and a learning rate of 1×10^{-4} with

Adam [15] optimizer to train these tasks, and save the parameters with the highest DSC or AUC score on validation sets for segmentation or classification tasks.

E.3.2 Implementation for fine-tuning evaluation We further take fine-tuning evaluation to evaluate the transferring ability thus demonstrating the great potential for initialization of downstream tasks. We most follow Models Genesis [22] for training fine-tuning models. 1) For segmentation tasks (SHC, SAC, SBM), we connect the whole backbone network $\mathcal{N}_\theta$ with a convolutional layer followed by a Softmax activation function, thus constructing a segmentation framework. The gradient optimizes the all parameters in this framework during the backpropagation. 2) For classification task (CCC), we use the encoder part of the backbone network, and the encoder is connected to a linear layer followed with a Sigmoid activation function. Like the segmentation tasks, the gradient optimizes all parameters in the framework. Like the implementation of linear evaluation, we also use the batch size of 1 and learning rate of 1×10^{-4} with Adam [15] optimizer, and save the parameters with the highest DSC or AUC score on validation sets.

References

[1] Octavian Andrei. *3D affine coordinate transformations*. 2006. 4

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, June 2021. 2

[5] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 3

[6] Yuhang Ding, Xin Yu, and Yi Yang. Modeling the probabilistic distribution of unlabeled data for one-shot medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1246–1254, 2021. 6

[7] Ramtin Gharleghi, Dona Adikari, Katy Ellenberger, Sze-Yuan Ooi, Chris Ellis, Chung-Ming Chen, Ruochen Gao, Yuting He, Raabid Hussain, Chia-Yen Lee, et al. Automated segmentation of normal and diseased coronary arteries-the asoca challenge. *Computerized Medical Imaging and Graphics*, page 102049, 2022. 5

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 1

[9] Greg Hamerly and Charles Elkan. Learning the k in k-means. *Advances in neural information processing systems*, 16, 2003. 2

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 3

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 3

[13] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2021. 2

[14] David N Kennedy, Christian Haselgrove, Steven M Hodge, Pallavi S Rane, Nikos Makris, and Jean A Frazier. Candishare: A resource for pediatric neuroimaging data. *Neuroinformatics*, 10(3):319, 2012. 6

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[16] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2

[17] Xiaoni Li, Yu Zhou, Yifei Zhang, Aoting Zhang, Wei Wang, Ning Jiang, Haiying Wu, and Weiping Wang. Dense semantic contrast for self-supervised visual representation learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1368–1376, 2021. 2

[18] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 1

[19] Marie-Pierre Revel, Samia Boussouar, Constance de Margerie-Mellon, Inès Saab, Thibaut Lapotre, Dominique Mompoint, Guillaume Chassagnon, Audrey Milon, Mathieu Lederlin, Souhail Bennani, et al. Study of thoracic ct in covid-19: The stoic project. *Radiology*, 301(1):E361–E370, 2021. 6

[20] Shuxin Wang, Shilei Cao, Dong Wei, Renzhen Wang, Kai Ma, Liansheng Wang, Deyu Meng, and Yefeng Zheng. Lt-net: Label transfer by learning reversible voxel-wise correspondence for one-shot medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 6

[21] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3

[22] Zongwei Zhou, Vatsal Sodha, Md Mahfuzur Rahman Siddiquee, Ruibin Feng, Nima Tajbakhsh, Michael B Gotway,

and Jianming Liang. Models genesis: Generic autodidactic models for 3d medical image analysis. In *International conference on medical image computing and computer-assisted intervention*, pages 384–393. Springer, 2019. 1, 3, 7

[23] Xiahai Zhuang, Lei Li, Christian Payer, Darko Štern, Martin Urschler, Mattias P Heinrich, Julien Oster, Chunliang Wang, Örjan Smedby, Cheng Bian, et al. Evaluation of algorithms for multi-modality whole heart segmentation: an open-access grand challenge. *Medical image analysis*, 58:101537, 2019. 5