Vector Contrastive Learning For *Pixel-Wise* Pre-Training In Medical Vision
(Supplementary Materials)

Contents

A. Theoretical Foundation

Existing pixel-wise contrastive learning (CL) still follows binary CL [4, 6–8, 10, 28, 34] that pulls positive pairs together and push negative pairs apart, being extremely limited by the over-dispersion problem, leading to a loss of local semantic continuity—an issue especially detrimental to medical image analysis. In contrast, we introduce a theoretical framework to derive the generalization bound for pixel-wise CL. We propose a novel CL paradigm, vector CL, which models the distances between pixel-wise features through a structured mapping function, controlling dispersion while preserving local correlations.

A.1. Recap of the Over-Dispersion Problem and Rademacher Complexity

Foundation of Contrastive Learning Let $x \in \mathcal{X}$ denote an input medical image, and assume that a network $f : \mathcal{X} \to \mathbb{R}^d$ extracts pixel-wise features $\{f(x)_i\}_{i=1}^N$ from the image, where $N$is the number of pixels. Define a similarity metric $\langle f_\theta(x_i), f_\theta(x_j)\rangle$ to measure the distance between pixel-wise features. In binary CL, the loss $\mathcal{L}_{BCL}$ for a given pixel on position i can be expressed as an InfoNCE-style [27] objective:

$$\mathcal{L}_{BCL} = -\log \frac{e^{\langle f(x)_i, f(x^+)_i\rangle/\tau}}{e^{\langle f(x)_i, f(x^+)_i\rangle/\tau} + \sum_j e^{\langle f(x)_i, f(x^-)_j\rangle/\tau}}, \quad (1)$$

where x^+ is a positive (similar) view of x and x^- represents negative samples; τ is a temperature parameter. For pixel-wise tasks, we additionally assume a local smoothness constraint: for any pixel on position i and its local neighborhood $\mathcal{N}(i)$,

$$\langle f(x)_i, f(x)_j\rangle \le \delta, \quad \forall j \in \mathcal{N}(i), \quad (2)$$

with δ being a small constant relative to the range of possible feature values Δ. The upper bound of the δ to describe the dispersion of the local features: for any two pixels on position j, k in the local neighborhood $\mathcal{N}(i)$,

$$\delta = \max_{j,k\in\mathcal{N}(i)} |\langle f(x)_i, f(x)_j\rangle - \langle f(x)_i, f(x)_k\rangle|. \quad (3)$$

Foundation of Rademacher Complexity Let $\mathcal{F}$ denote the hypothesis space of feature extractors. The empirical Rademacher complexity [26] is defined as

$$\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{f\in\mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(x_i) \right], \quad (4)$$

where $\{\sigma_i\}$ are independent Rademacher variables. For pixel-wise learning, one might consider nN samples (with n images and N pixels per image). However, the local

smoothness constraint effectively reduces the "degrees of freedom" in each local neighborhood [1, 22]. We can thus define an effective local Rademacher complexity as

$$\mathfrak{R}_{n,\text{local}}(\mathcal{F}) \leq \left(\frac{\delta}{\Delta}\right) \mathfrak{R}_n(\mathcal{F}), \tag{5}$$

where the factor δ/Δ represents the reduction in complexity due to local smoothness.

By standard generalization theory [26], for any $f \in \mathcal{F}$, with high probability (at least $1-\varepsilon$), the generalization error satisfies

$$R(f) \leq \hat{R}(f) + 2\mathfrak{R}_{n,\text{local}}(\mathcal{F}) + O\left(\sqrt{\frac{\log(1/\varepsilon)}{nN}}\right), \tag{6}$$

or equivalently,

$$R(f) \leq \hat{R}(f) + 2\left(\frac{\delta}{\Delta}\right) \mathfrak{R}_n(\mathcal{F}) + O\left(\sqrt{\frac{\log(1/\varepsilon)}{nN}}\right). \tag{7}$$

Thus, reducing the maximum local difference δ through better modeling of local continuity directly tightens the generalization bound.

Over-Dispersion Problem In the binary CL, since there is no explicit control over how close or dispersed feature representations are within a local neighborhood $\mathcal{N}(i)$, the difference between adjacent pixel features δ_{BCL} is unrestricted and can increase without bound. Therefore, the features will be free to spread apart as long as they satisfy the binary CL objective. This can lead to *over-dispersion*, where nearby pixels develop large feature differences, even if they belong to the same structure or object in the image. Mathematically, this means that for neighboring pixels $j \in \mathcal{N}(i)$, their feature similarity can vary widely:

$$\langle f(x)_i, f(x)_j \rangle \rightarrow \text{anywhere within} [-\Delta, \Delta], \tag{8}$$

since no term in binary CL enforces a small difference. This means binary CL lacks a mechanism to prevent large local variations, the feature differences δ_{BCL} can span the entire range of possible feature values Δ. Thus, in the worst case:

$$\delta_{BCL} \approx \Delta. \tag{9}$$

This means adjacent pixel features could be as different as features from completely unrelated image regions, which is undesirable for pixel-wise tasks like segmentation or medical image analysis. Therefore, the generalization error satisfies in the binary CL is

$$R_{BCL}(f) \leq \hat{R}(f) + 2\mathfrak{R}_n(\mathcal{F}) + O\left(\sqrt{\frac{\log(1/\varepsilon)}{nN}}\right), \tag{10}$$

causing the over-dispersion problem and limiting the generalization.

A.2. Vector Contrastive Learning

Derivation for Vector Contrastive Learning Vector CL introduces an additional regression loss that explicitly enforces the mapping between feature distances and corresponding spatial displacement vectors. Its loss is

$$\mathcal{L}_{\text{vec}} = \|v - \mathcal{V}(d')\|, \tag{11}$$

where v represents the ground-truth displacement vector and $\mathcal{V}(d')$ is a mapping function applied on the computed feature distances d'. It is formulated as a weighted sum of a vector template matrix $\mathbb{V}$, i.e., $\mathcal{V}(d') = \sum_{j=0}^{\mathcal{N}(i)} \mathbb{V}^j d'^j$, where the $d'^j = \frac{e^{\langle f(x)_i, f(x)_j \rangle/\tau}}{\sum_j^{\mathcal{N}(i)} e^{\langle f(x)_i, f(x)_j \rangle/\tau}}$. Therefore, the loss in vector CL $\mathcal{L}_{VCL}$ can be further formulated as

$$\mathcal{L}_{VCL} = \left\| v - \sum_{j=0}^{\mathcal{N}(i)} \mathbb{V}^j \frac{e^{\langle f(x)_i, f(x)_j \rangle/\tau}}{\sum_j^{\mathcal{N}(i)} e^{\langle f(x)_i \cdot f(x)_j \rangle/\tau}} \right\| \tag{12}$$

To minimize the $\mathcal{L}_{VCL}$, the model must adjust feature similarity $\langle f(x)_i, f(x)_j \rangle$ so that the weighted sum of $\mathbb{V}^j$ matches v. We assume that the v can be modeled as

$$v = \sum_j^{\mathcal{N}(i)} \alpha_j \mathbb{V}^j \ (\alpha_j \geq 0, \sum_j^{\mathcal{N}(i)} \alpha_j = 1), \tag{13}$$

the model enforces $\frac{e^{\langle f(x)_i, f(x)_j \rangle/\tau}}{Z} \approx \alpha_j$, where the $Z = \sum_j^{\mathcal{N}(i)} e^{\langle f(x)_i, f(x)_j \rangle}$. Therefore, the similarity $\langle f(x_i), f(x_j) \rangle$ can be further approximated as

$$\langle f(x_i), f(x_j) \rangle \approx \tau \log \alpha_j + \tau \log Z. \tag{14}$$

The maximum difference of our vector CL between any two pixels in the $\mathcal{N}(i)$ can be formulated is:

$$\begin{aligned} \delta_{VCL} &= \max_{i,k} |\langle f(x_i), f(x_j) \rangle - \langle f(x_i), f(x_k) \rangle| \\ &= \max_{i,k} |\tau \log \alpha_j + \tau \log Z - \tau \log \alpha_k - \tau \log Z| \\ &= \tau \max_{i,k} |\log \frac{\alpha_j}{\alpha_k}|. \end{aligned} \tag{15}$$

Owing to the $\alpha_j, \alpha_k \in [\alpha_{\min}, 1]$ ($\alpha_{\min} > 0$), when $\alpha_j = 1, \alpha(k) = \alpha_{\min}$ the δ_{VCL} will be:

$$\delta_{VCL} \leq \tau \log \frac{1}{\alpha_{\min}}. \tag{16}$$

due to the normalization of weights ($\sum_j \alpha_j = 1$), only when $\alpha_j = 1$, all other $\alpha \in \mathcal{N}(i)$ are 0. This situation where one weight is exactly 1 and all others are zero is degenerate (and would essentially revert to binary CL), so typically the effective δ_{VCL} is much smaller than the worst-case bound, i.e., $\delta_{VCL} \leq \tau \log \frac{1}{\alpha_{\min}} \ll \Delta$.

Generalization Bound for Vector Contrastive Learning Recalling the generalization error bound for any $f \in \mathcal{F}$ in Equ.7, we substitute the improved local dispersion bound δ_{VCL} for δ in the vector CL case and use the fact in Equ.16, so that we will obtain:

$$R_{VCL}(f) \leq \hat{R}(f) + 2\left(\frac{\tau \log \frac{1}{\alpha_{\min}}}{\Delta}\right)\mathfrak{R}_n(\mathcal{F}) + O\left(\sqrt{\frac{\log(1/\varepsilon)}{nN}}\right) \tag{17}$$

Since typically $\tau \log(1/\alpha_{\min}) \ll \Delta$ (because the normalization forces the weights to be spread across several pixels rather than concentrating on one), the factor $\frac{\tau \log \frac{1}{\alpha_{\min}}}{\Delta}$ is much smaller than 1. In contrast, in the binary CL setting, the worst-case dispersion is $\delta_{BCL} \approx \Delta$, leading to a much looser bound. Compared with the Equ.10, our vector CL making a tighter generalization bound.

B. Technical Details

B.1. Appearance transformation operation $\mathcal{T}_{ap}$

The appearance transformation operation $\mathcal{T}_{ap}$ consists of random noise, blurring, contrast, brightness, and in-painting. Specifically,

1. For the random noise operation, it randomly generates Gaussian noise from the Gaussian distribution with $\mu = 0$ (mean) and $\sigma \in [0, 0.02]$ (variance), and adds it to the image.
2. For the random blurring, it performs random Gaussian blurring with $\mu = 0$ and $\sigma \in [0, 0.05]$ on the image.
3. For the random contrast, it performs multiplicative transformation with $(x - \bar{x}) * \gamma + \bar{x}$ on the images x, where the $\bar{x}$ is the average value and the $\gamma \in [0.5, 1.5]$ is the scaling value.
4. For the random brightness, it performs additive transformation with $x + \beta$ on the images, where the β is randomly sampled from a Gaussian distribution with $\mu = 0$ and $\sigma \in [0, 0.1]$.
5. For the random in-painting, it randomly selects boxes, and the contents of these regions are replaced by the noise from a uniform distribution.

These five sub-operations are used sequentially with a probability of 0.9 for the image with appearance transformation.

B.2. Space transformation operation $\mathcal{T}_{sp}$

We utilize random affine transformation [2] to construct the space transformation operation $\mathcal{T}_{sp}$. For a clearer illustration, we introduce it in 2D situation. As shown in Equ.18, it as four sub-operations including the translation $t_x, t_y \in [-0.2, 0.2]$, rotation $\theta \in [-\pi/9, \pi/9]$, shearing $sh_x, sh_y \in [-\pi/32, \pi/32]$, and scaling $s_x, s_y \in [0.5, 1.5]$, where the H, W in the Equ.18 are the height and width of the medical images. These operations are These operations form an affine transformation matrix ϕ_{ab} that transforms the position of each pixel $\mathbf{p}_a$ in the image to a new position $\mathbf{p}_{ab}$. Therefore, the ground truth vector v^i_{ab} will be generated via the coordinate difference on the image grid $v^i_{ab} = \mathbf{p}^i_{ab} - \mathbf{p}^i_a$, and the vectors at each position jointly construct the displacement vector field (DVF) $\psi_{ab} = \{v^i_{ab}\}_{i \in \mathbb{R}}$. The space-transformed view x_b is generated by the DVF ψ_{ab} via moving the pixels to target positions and completing the non-integer coordinates by bilinear interpolation.

B.3. Mask ϵ_{ab} indicates matched regions

The mask ϵ_{ab} eliminates the content mismatch of two views x_a, x_b caused by the spatial transformation. It is calculated from the DVF ψ_{ab} according to whether the transformed coordinates exceed the image grid. For each position i on image grid, the value of the ϵ^i_{ab} is

$$\epsilon^i_{ab} = \mathbf{1}_{(0,H)}(v^i_x + p^i_x) \wedge \mathbf{1}_{0,W}(v^i_y + p^i_y), \tag{19}$$

where the $\mathbf{1}_{(0,H)}(\cdot)$ and $\mathbf{1}_{(0,W)}(\cdot)$ are Iverson Notation [17]. Therefore, the mask ϵ_{ab} be generated whose 0 values indicate the mismatched regions and 1 values indicate the matched regions.

B.4. Multi-scale fusion operation $\bigodot$ in our VPA

The fusion operation $\bigodot$ in our vector pyramid aggregation integrates the DVFs from different levels. Here, we utilize the fusion of DVFs in level 0 ψ'^0_{ab} and level 1 ψ'^1_{ab} to introduce this operation. It has three steps: **1) Scale alignment:** The level 0 DVF ψ'^0_{ab} is up-sampled to the same size of level 1 DVF ψ'^1_{ab} via bilinear interpolation. Then the values are enlarged (double) to adapt to the size of the level 1 grid. **2) Space alignment:** To align the center coordinates of the vectors in these two level DVFs, the level 0 DVF is transformed to align the level 1 DVF. **3) Vector fusion:** Finally, the DVFs in two levels are fused via addition. The whole fusion operation $\bigodot$ is formulated as,

$$\psi'^1_{ab} \bigodot \psi'^0_{ab} = \overbrace{\psi'^1_{ab} + \underbrace{\psi'^1_{ab}(\overbrace{2 * \mathcal{I}_{H\times W}(\psi'^0_{ab})}^{\text{Scale alignment}})}_{\text{Space alignment}}}^{\text{Vector fusion}}, \tag{20}$$

where H and W are the height and width of level 1 grid, and $\mathcal{I}$ is the bilinear interpolation.

B.5. Vector template $\mathbb{V}^{N\times N}$ maps distances as vector

The vector template matrix describes the basic spatial relationship of pixels that indicates the vectors in which the center coordinates pointing to the coordinates in the receptive field. As shown in Fig.1, for a clear illustration, we assume that the template is a 2D matrix (2 channels) with 3×3 receptive fields. The value in each position is the vector, i.e., (x, y), that indicates the relative position offset from the center coordinate so that the $\mathbb{V}^{3\times3} = \{(x, y)|x, y \in$

$$\phi_{ab} = \overbrace{\begin{bmatrix} 1 & 0 & t_x H \\ 0 & 1 & t_y W \\ 0 & 0 & 1 \end{bmatrix}}^{\text{Translation}} \overbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\text{Rotation}} \overbrace{\begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\text{Shearing}} \overbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\text{Scaling}} = \begin{bmatrix} a_{11} & a_{12} & t_x H \\ a_{21} & a_{22} & t_y W \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} s_x \cos\theta + sh_x s_y \sin\theta & -s_x \sin\theta + sh_x s_y \cos\theta & t_x H \\ sh_y s_x \cos\theta & -sh_y s_x \sin\theta + s_y \cos\theta & t_y W \\ 0 & 0 & 1 \end{bmatrix}, \tag{18}$$
$$v_{ab}^i = \mathbf{p}_{ab}^i - \mathbf{p}_a^i = \phi_{ab} \cdot \begin{bmatrix} p_x^i \\ p_y^i \\ 1 \end{bmatrix} - \begin{bmatrix} p_x^i \\ p_y^i \\ 1 \end{bmatrix} = \begin{bmatrix} (a_{11}-1)p_x^i + a_{12}p_y^i + t_x H \\ a_{21}p_x^i + (a_{22}-1)p_y^i + t_y W \\ 1 \end{bmatrix} = \begin{bmatrix} v_x^i \\ v_y^i \\ 1 \end{bmatrix}, \psi_{ab} = \{v_{ab}^i\}_{i \in \mathbb{R}}$$

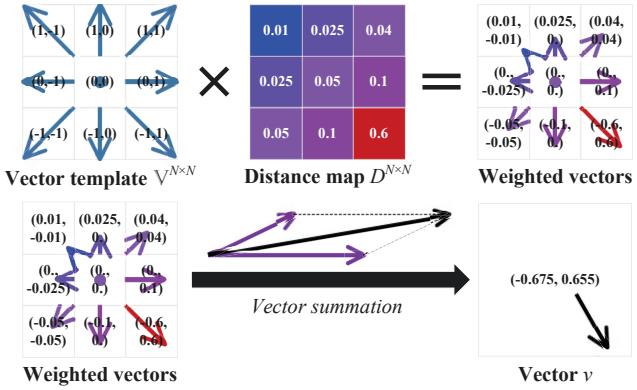


Figure 1. The details of our vector template matrix. It is multiplied to the distance map mapping the distances as vectors. The values in the distance map are examples.

Dataset	Type	Num	D	P	Task
CANDI [19]	3D T1 brain MRI	103	√		S
FeTA21 [30]	3D T2 brain MRI	80	√		S
SCR [35]	2D chest X-ray	247	√		S
KiPA22 [12]	3D kidney CT	130	√		S
FIVES [18]	2D fundus	800	√		S
PDCXR [20]	2D chest X-ray	5,956	√		C
STOIC [31]	3D chest CT	2,000	√		C
ChestX-ray8 [36]	2D chest X-ray	112,120		√	-
PPMI (T1) [25]	3D T1 brain MRI	837		√	-

Table 1. A Total of 9 publicly available datasets are involved in this paper for the experiments, achieving great reproducibility. The "D" and "P" mean the datasets are used for downstream tasks and pretraining tasks. The "S" and "C" are the segmentation and classification tasks.

$\{-1, 0, 1\}\}$. To map the distances as vectors, the vector template matrix is multiplied by the distance map $D^{N\times N}$ of the receptive field corresponding to the center coordinate, i.e., $D^{3\times3} \cdot \mathbb{V}^{3\times3} = \{(x, y)|x = \mathbb{V}_{[0]}^{i,j} * D^{i,j}, y = \mathbb{V}_{[1]}^{i,j} * D^{i,j}, i, j \in \{0, 1, 2\}\}$, for weighted vectors in a matrix. Finally, these weighted vectors are summed for the vector $v = \sum_{i,j}^{j,j\in\{0,1,2\}} D^{i,j} \cdot \mathbb{V}^{i,j}$ indicating the correspondence between the center coordinate $(0, 0)$ and the coordinates in the field.

C. Experiment Details

C.1. Datasets

As shown in Tab.1, eight publicly available datasets are involved in this paper, specifically,

CANDI [19] The Child and Adolescent NeuroDevelopment Initiative (CANDI) dataset has 103 T1 brain MR volumes from 57 males and 46 females. Totally 28 brain tissue regions are annotated for masks. For the segmentation task (CANDI_S), 40, 20, and 43 volumes are used as training, validation, and test sets. Following [13], we resize and crop $160\times160\times128$ volumes on the brain regions, and then normalize the intensity via $\frac{x-\min(x)}{\max(x)-\min(x)}$.

FeTA21 [30] The Fetal Tissue Annotation 2021 (FeTA21) challenge dataset has 120 fetal T2w brain MR volumes, and 80 of them are available as the training data in the challenge. We split the 80 volumes and set 20 of them as the training set, 20 of them as the validation set, and 40 of them as test set. We normalize the intensity of the images via $\frac{x-\min(x)}{\max(x)-\min(x)}$ and use the $128 \times 128 \times 128$ random cropping to unify the input size.

SCR [35] The segmentation of chest radiographs (SCR) dataset is from the JSRT database [33] with 247 2048×2048 posterior to anterior (PA) chest radiographs. Three chest-related structures, including the heart, chest, and clavicle, are annotated for masks. We set 100 of them as the training set, 47 of them as a validation set, and 100 of them as the test set. Following [13], we resize the images to 512×512, and normalize the intensity via $x/255$ for the segmentation task (SCR_S).

KiPA22 [12] The kidney parsing 2022 (KiPA22) challenge dataset has 130 kidney CT volumes. These images are cropped from 130 abdominal CT angiography volumes for kidney regions with tumors. Four kidney-related struc-

tures, including the kidney, vessel, vein, and tumor, are annotated for masks. In 2D evaluation (KiPA$^{2D}_S$), 13,846 2D slices from 70 volumes are used as the training set, 5,864 2D slices from 30 volumes are used as the validation set, 5,959 2D slices from 30 volumes are used as the test set. We normalize the intensity of the images via $\frac{\max(\min(0,x),2048)}{2048}$, and use the 128×128 randomly to unify the input size. For 3D evaluation (KiPA22$^{3D}_S$), 70, 30, and 30 volumes are used as training, validation, and test sets. We normalize the intensity of the images via $\frac{\max(\min(0,x),2048)}{2048}$, and use the $128 \times 128 \times 128$ randomly to unify the input size.

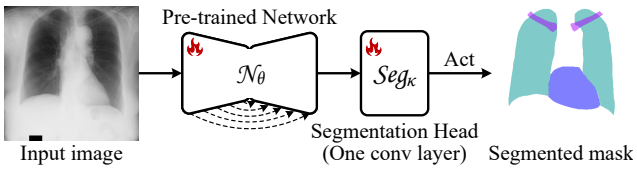
FIVES [18] The fundus image vessel segmentation (FIVES) dataset consists of 800 color fundus photographs with vessel annotation from 573 patients. 540, 60, and 200 of the images are used as training, validation, and test sets. We resize the images to 512×512, and normalize the intensity via $x/255$ for the segmentation task (FIVES$_S$).

PDCXR [20] The pneumonia Detection Using Chest X-ray (PDCXR) dataset has 5,856 chest X-ray images for the diagnosis of pneumonia. Following [20], 3,659 of them are used as training set (2,714 pneumonia, 945 normal), 1,573 of them are used as validation set (1,169 pneumonia, 404 normal), and 624 of them are used as test set (390 pneumonia, 234 normal). We resize the images to 512×512, and normalize the intensity via $x/255$ for classification task (PDCXR$_C$).

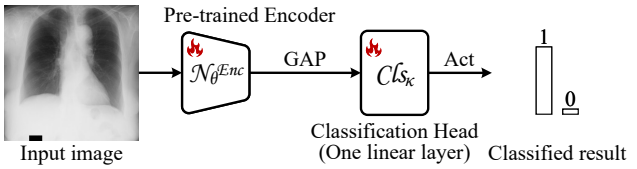
STOIC [31] The STOIC dataset is from the STOIC 2021 challenge with 2000 chest CT volumes for COVID-19 diagnosis. Following [14], 1000 of them are used as training set (603 COVID-19, 397 normal), 400 of them are used as validation set (241 COVID-19, 159 normal), and 600 of them are used as test set (361 COVID-19, 239 normal). We utilize the Lungmask [15] to extract the lung regions avoiding the interference of the background, resample the resolutions to $1mm^3$, and normalize the intensity via $\frac{\max(\min(0,x),2048)}{2048}$ for classification task (STOIC$_C$).

ChestX-ray8 [36] The ChestX-ray8 is our pre-training dataset in 2D evaluation. It has 112,120 frontal-view chest X-ray images with 1024×1024 resolution. 44,810 of them are scanned from the anterior to posterior (AP) view and 67,310 of them are scanned from the PA view. We resize the images to 512×512, and normalize the intensity $x/255$. During the pre-training, 384×384 patches are randomly cropped for augmentation.

PPMI [25] The PPMI is our pre-training dataset in 3D evaluation. It is extracted from the PPMI database which is a large Parkinson progression marker initiative database, for 837 T1 brain MR volumes. Following the pre-processing in the CANIA dataset, we resize and crop 160×160×128 volumes on the brain regions, and then normalize the intensity via $\frac{x-\min(x)}{\max(x)-\min(x)}$. We also extract the brain regions via HD-BET [16] to avoid the interference of background. During the pre-training, 128×128×128 patches are randomly cropped for augmentation.



a) Fine-tuning in downstream ***segmentation*** tasks



b) Fine-tuning in downstream ***classification*** tasks

Figure 2. The detailed implementations in our downstream tasks, including the a) segmentation and b) classification.

C.2. Implementations

Our experiments are implemented by PyTorch [29] which is a widely recognized deep learning library on NVIDIA A100 SXUM4 GPU with 40 GB memory. Specifically,

C.2.1. Pre-training implementation

In both 2D evaluation and 3D evaluation, we utilize the Adam optimizer [21] with the learning rate of 1×10^{-4} and iterations of 2×10^5. For 2D evaluation, 24 2D chest X-ray images from ChestX-ray8 [36] dataset are randomly sampled as a batch in each iteration with 384×384 random cropping augmentation. For 3D evaluation, 2 3D brain MR volumes from the PPMI [25] dataset are randomly sampled as a batch in each iteration with 128×128×128 random cropping augmentation.

C.2.2. Downstream implementation

As shown in Fig.2, following [14], we utilize the fine-tuning evaluation to demonstrate the adaptation ability in downstream tasks. The gradient optimizes all parameters through the frameworks during the training. All the downstream tasks are trained by Adam optimizer [21] with the learning rate of 1×10^{-4} and iterations of 4×10^4.

For segmentation tasks, the whole pre-trained network $\mathcal{N}_\theta$ is utilized to extract the pixel-wise features (level 4) f^4, and these features are putted into a new segmentation head Seg_κ to predict the final segmentation masks. The segmentation head is a convolutional layer to map the features to the channels of segmentation targets, thus constructing a segmentation framework for the SCR$_S$, KiPA22$^{2D}_S$, FIVES$_S$, CANDI$_S$, and KiPA22$^{3D}_S$ tasks. We use the sum of Dice loss [37] and cross-entropy loss [24] between the predicted masks and ground truths to train the segmentation tasks.

For classification tasks, the encoder part of the pre-trained network $\mathcal{N}^{Enc}_\theta$ is utilized to extract the high-level

Backbone	Methods	$\text{SCR}_S^{25\%}$	PDCXR_C	KiPA22_S^{2D}	FIVES_S	AVG
U-Net [32]	Scratch	81.8	90.4	74.1	79.4	81.4
	COVER	94.0 (+12.2)	95.9 (+5.5)	80.0 (+5.9)	87.2 (+7.8)	89.3 (+7.9)
TransUNet [5]	Scratch	89.2	77.4	57.6	81.7	76.5
	COVER	93.4 (+4.2)	90.6 (+13.2)	76.7 (+19.1)	86.5 (+4.8)	86.8 (+10.3)
SwinUNet [3]	Scratch	85.6	93.0	69.1	77.0	81.2
	COVER	86.9 (+1.3)	95.6 (+2.6)	72.6 (+3.5)	84.8 (+7.8)	85.0 (+3.8)
U-KAN [23]	Scratch	89.4	89.3	61.3	80.3	80.1
	COVER	93.8 (+4.4)	95.6 (+6.3)	71.8 (+10.5)	84.9 (+4.6)	86.5 (+6.4)

Table 2. Our COVER has great cross-architecture compatibility that achieves significant improvement on all U-Net, TransUNet, SwinUNet, and UKAN. We evaluate it in our 2D setting.

$\text{SCR}_S^{25\%}$		PDCXR_C		KiPA22_S^{2D}		FIVES_S		AVG	
Cor ↑	p ↓	*Cor* ↑	p ↓	*Cor* ↑	p ↓	*Cor* ↑	p ↓	*Cor* ↑	p ↓
0.908	<0.001	0.898	< 0.001	0.949	<0.001	0.995	<0.001	0.938	< 0.001

Table 3. The test-retest reliability analysis [11] of our COVER on the tasks of 2D evaluation. The *Cor* is the Pearson correlation coefficient [9], and the p is the p-value.

features (level 0) f^0, and a global adaptive pooling is used to compress features. Then these compressed features are putted into a classification head Cls_κ to achieve the classified results, thus constructing a segmentation framework for the PDCXR_C and STOIC_C tasks. We utilize the cross-entropy loss [24] between the predicted categories and ground truths to train the classification.

D. More Framework Analysis and Results

D.1. Analysis of the reliability

As shown in Tab.3, we utilize the test-retest analysis [11] to evaluate the reliability of our COVER. We pre-trained and adapted our method twice in our 2D evaluation from different initialization states, and then calculated the correlation coefficient [9] and p-value between these two results. Our COVER achieved 0.938 average Cors over four tasks demonstrating very high consistency between two training sessions. All p-values lower than 0.001 illustrated the significant consistency. Therefore, these results show our powerful reliability across initialization states, supporting the implementation in the application.

D.2. Analysis of cross-architecture compatibility

As shown in Tab.2, our COVER has great cross-architecture compatibility that achieves significant improvement on all U-Net [32] (CNN-based), TransUNet [5] (CNN-Transformer-based), SwinUNet [3] (Transformer-based), and UKAN [23] (KAN-based). We utilize these four networks with different paradigms as the backbone network in our COVER framework and train the framework on our 2D evaluation setting. Compared with the "scratch" on these networks, our COVER has achieved more than 3% average improvement owing to the learned knowledge from the pre-training data. Especially, on the TransUNet which utilizes a vision transformer and is easy to fall into an over-fitting

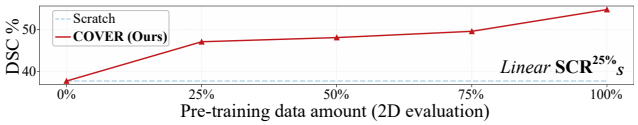


Figure 3. The analysis of the pre-training data amount. When enlarging the pre-training data amount, the performance of our COVER is improved gradually.

state, our COVER brings a significant 10.3% average improvement.

D.3. Analysis pre-training data amount

As shown in Fig.3, we evaluate the variation of our COVER's performance with the enlarging of the pre-training data amount on our 2D evaluation and adapt the pre-trained model to the $\text{SCR}_S^{25\%}$ via linear evaluation. Compared with the "scratch", our COVER will bring a significant improvement even though only 25% pre-training data is involved. When further enlarging the -pre-training dataset, the gain of performance gradually decreases owing to the similarity of the same category of medical images in the pre-training dataset. Fortunately, our COVER has a powerful modeling ability for pixel-wise features, enabling the learning from the details effectively, so the performance of the model is still improving gradually.

D.4. Visualization of the segmentation results

The visualization of the segmentation results (Fig.4) demonstrates our superiority in the adaptation of pixel-wise tasks. Due to our vector CL with distance modeling, the pre-training enables the networks pixel-wise representation with controllable dispersion. It has two observations: 1) For large objects with varied appearances like the tumors in KiPA22_S^{2D} and KiPA22_S^{3D}, our COVER achieves excellent integrity owing to its disentanglement of underlying explanatory factors hidden in low-level sensory data. The BYOL, SimSiam, Model Genesis, Rotation, and "Scratch" have poor performance on the tumors, because of their lack of modeling for distinct features. 2) For small objects like the thin vessels in FIVE_S, small brain tissues in CANDI_S and FeTA21 $_S$, and clavicles in $\text{SCR}_S^{25\%}$, our COVER also has fine segmentation owing to our pixel-wise representation with distance modeling. Such representation preserves the distinction of detail features thus making the networks easy to segment these regions in downstream tasks.

D.5. Visualization of multi-scale vectors

We visualize more results of our regressed multi-scale vectors (corresponding to Fig.8 in the manuscript) in Fig.5. It aligns the appearance-transformed view x_a to the space-transformed view x_b via the deformation of regressed vectors in different scale levels. In a large number of cases,

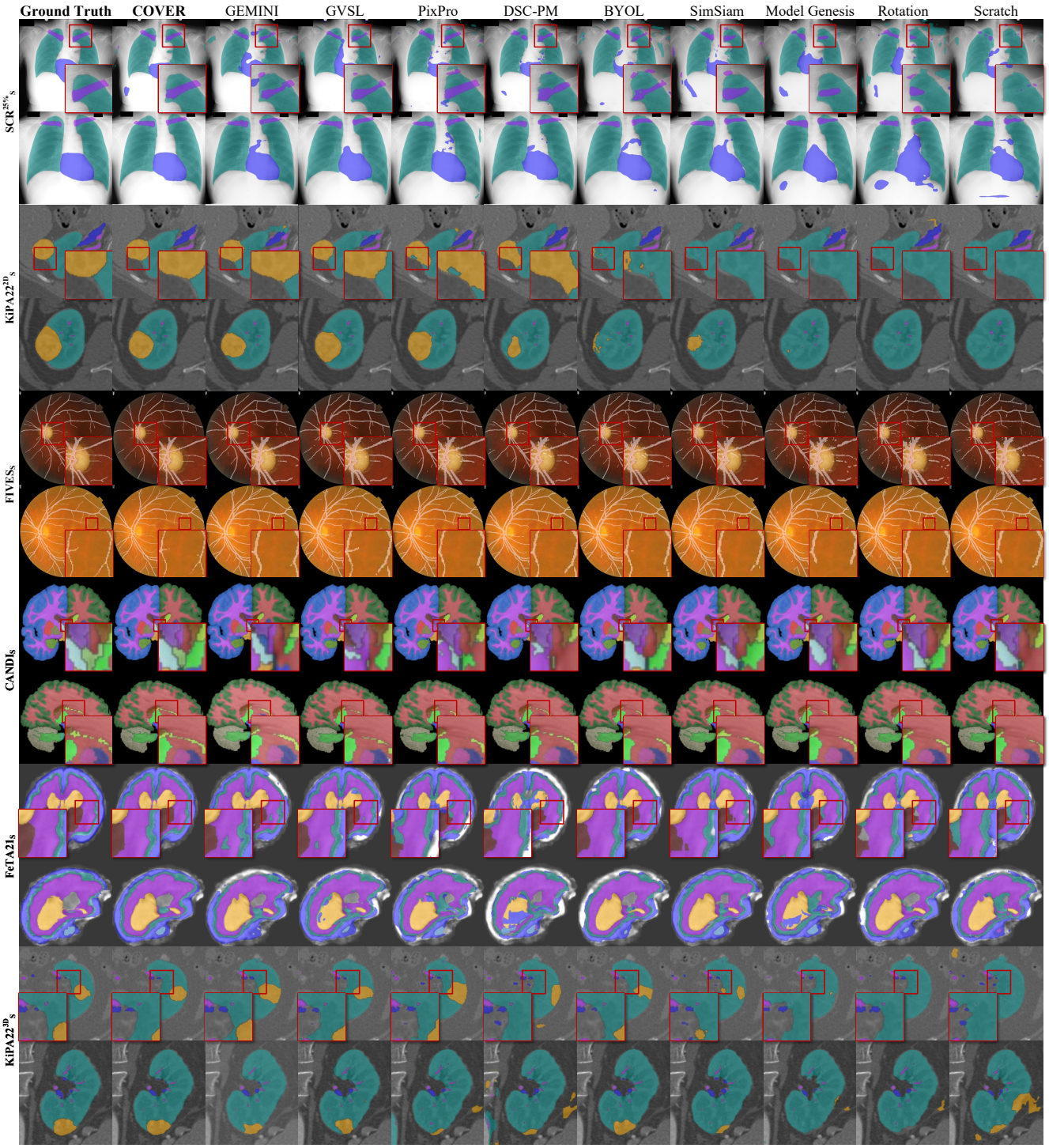


Figure 4. The visualization of the segmentation results for the methods with the top 2 average scores in each type. Our COVER achieves excellent integrity and fine segmentation ability for thin structures owing to its effective representation of pixel-wise features.

we can find that our VPA effectively aligns the regions with the same semantics between the views, illustrating that the regressed vectors are able to indicate the correspondence of the same semantics. Therefore, this means that in the embedding space, our vector CL can model closer distances between the same semantic objects and farther distances between different semantic objects, thus enabling our COVER to discover accurate correspondence in the receptive field.

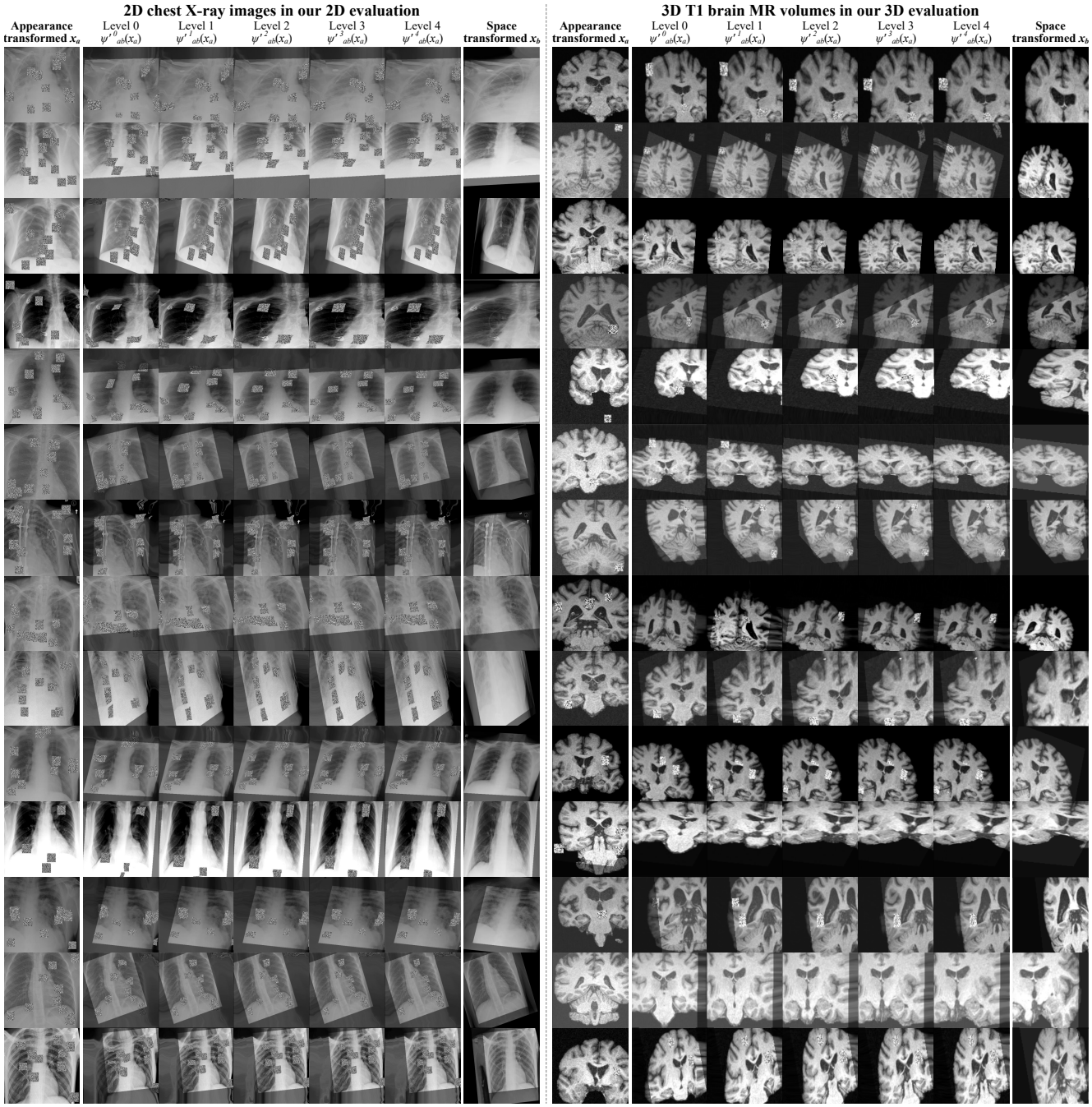


Figure 5. **Extension of multi-scale vectors analysis** (Fig.8 in manuscript): The visualization of the regressed vectors via the alignment of two views. Our VPA is able to discover the correspondence of the semantics at multiple levels for both global and detailed representation.

References

[1] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002. 2

[2] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992. 3

[3] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022. 6

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and

Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018. 1

[5] Jieneng Chen, Jieru Mei, Xianhang Li, Yongyi Lu, Qihang Yu, Qingyue Wei, Xiangde Luo, Yutong Xie, Ehsan Adeli, Yan Wang, et al. Transunet: Rethinking the u-net architecture design for medical image segmentation through the lens of transformers. *Medical Image Analysis*, 97:103280, 2024. 6

[6] 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. 1

[7] 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, 2021.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1

[9] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009. 6

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Pires, Zhaohan Guo, Mohammad Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Neural Information Processing Systems*, 2020. 1

[11] Louis Guttman. A basis for analyzing test-retest reliability. *Psychometrika*, 10(4):255–282, 1945. 6

[12] Yuting He, Guanyu Yang, Jian Yang, Rongjun Ge, Youyong Kong, Xiaomei Zhu, Shaobo Zhang, Pengfei Shao, Huazhong Shu, Jean-Louis Dillenseger, et al. Meta grayscale adaptive network for 3d integrated renal structures segmentation. *Medical Image Analysis*, 71:102055, 2021. 4

[13] Yuting He, Rongjun Ge, Xiaoming Qi, Yang Chen, Jiasong Wu, Jean-Louis Coatrieux, Guanyu Yang, and Shuo Li. Learning better registration to learn better few-shot medical image segmentation: Authenticity, diversity, and robustness. *IEEE Transactions on Neural Networks and Learning Systems*, 35(2):2588–2601, 2022. 4

[14] Yuting He, Guanyu Yang, Rongjun Ge, Yang Chen, Jean-Louis Coatrieux, Boyu Wang, and Shuo Li. Geometric visual similarity learning in 3d medical image self-supervised pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[15] Johannes Hofmanninger, Forian Prayer, Jeanny Pan, Sebastian Röhrich, Helmut Prosch, and Georg Langs. Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem. *European Radiology Experimental*, 4:1–13, 2020. 5

[16] Fabian Isensee, Marianne Schell, Irada Pflueger, Gianluca Brugnara, David Bonekamp, Ulf Neuberger, Antje Wick, Heinz-Peter Schlemmer, Sabine Heiland, Wolfgang Wick, et al. Automated brain extraction of multisequence mri using artificial neural networks. *Human brain mapping*, 40(17): 4952–4964, 2019. 5

[17] Kenneth E Iverson. A programming language. In *Proceedings of the May 1-3, 1962, spring joint computer conference*, pages 345–351, 1962. 3

[18] Kai Jin, Xingru Huang, Jingxing Zhou, Yunxiang Li, Yan Yan, Yibao Sun, Qianni Zhang, Yaqi Wang, and Juan Ye. Fives: A fundus image dataset for artificial intelligence based vessel segmentation. *Scientific data*, 9(1):475, 2022. 4, 5

[19] 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. 4

[20] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 172(5):1122–1131, 2018. 4, 5

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] Vladimir Koltchinskii. Local rademacher complexities and oracle inequalities in risk minimization. 2006. 2

[23] Chenxin Li, Xinyu Liu, Wuyang Li, Cheng Wang, Hengyu Liu, and Yixuan Yuan. U-kan makes strong backbone for medical image segmentation and generation. *arXiv preprint*, 2024. 6

[24] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International conference on Machine learning*, pages 23803–23828. PMLR, 2023. 5, 6

[25] Kenneth Marek, Danna Jennings, Shirley Lasch, Andrew Siderowf, Caroline Tanner, Tanya Simuni, Chris Coffey, Karl Kieburtz, Emily Flagg, Sohini Chowdhury, et al. The parkinson progression marker initiative (ppmi). *Progress in neurobiology*, 95(4):629–635, 2011. 4, 5

[26] Mehryar Mohri. Foundations of machine learning, 2018. 1, 2

[27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pages 1–31, 2024. 1

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[30] Kelly Payette, Hongwei Bran Li, Priscille de Dumast, Roxane Licandro, Hui Ji, Md Mahfuzur Rahman Siddiquee, Daguang Xu, Andriy Myronenko, Hao Liu, Yuchen Pei, et al.

Fetal brain tissue annotation and segmentation challenge results. *Medical image analysis*, 88:102833, 2023. 4

[31] 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. 4, 5

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[33] Junji Shiraishi, Shigehiko Katsuragawa, Junpei Ikezoe, Tsuneo Matsumoto, Takeshi Kobayashi, Ken-ichi Komatsu, Mitate Matsui, Hiroshi Fujita, Yoshie Kodera, and Kunio Doi. Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *American Journal of Roentgenology*, 174 (1):71–74, 2000. 4

[34] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020. 1

[35] Bram Van Ginneken, Mikkel B Stegmann, and Marco Loog. Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database. *Medical image analysis*, 10(1):19–40, 2006. 4

[36] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017. 4, 5

[37] Rongjian Zhao, Buyue Qian, Xianli Zhang, Yang Li, Rong Wei, Yang Liu, and Yinggang Pan. Rethinking dice loss for medical image segmentation. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 851–860. IEEE, 2020. 5