Appendix for *Residual Aligner-based Network (RAN): Motion-Aware Structure for Coarse-to-fine Deformable Image Registration*

Jian-Qing Zheng[1,2], Ziyang Wang[3], Baoru Huang[4], Ngee Han Lim[1], and Bartłomiej W. Papież[2,5]

[1] The Kennedy Institute of Rheumatology, University of Oxford, U.K.
[2] Big Data Institute, University of Oxford, U.K.
[3] Department of Computer Science, University of Oxford, U.K.
[4] Department of Surgery and Cancer, Imperial College London
[5] Nuffield Department of Population Health, University of Oxford, UK
{jianqing.zheng@kennedy,bartlomiej.papiez@bdi}.ox.ac.uk

1 Math

The denotations of symbols in this supplementary material are shown in the Tab. 1, which are the same as in the manuscript.

1.1 Proof of Regional Dependency

The pooling mapping $\mathcal{P}$ from the original full-resolution image's coordinate $\boldsymbol{x}$ to the k_{th} feature map with pool size p_k is denoted as:

$$\mathcal{P}(\boldsymbol{x}; p_k) := \lfloor \boldsymbol{x}/p_k \rfloor, \tag{1}$$

which thus satisfy:

$$\exists\ (\boldsymbol{x},\boldsymbol{y}) \in \{(\boldsymbol{x},\boldsymbol{y}) | \|\boldsymbol{x}-\boldsymbol{y}\|_\infty < p_k\},\ \mathcal{P}(\boldsymbol{x}; p_k) = \mathcal{P}(\boldsymbol{y}; p_k) \tag{2}$$

As stated in the manuscript, coarse-to-fine registration implies $\forall k \in [1, K) \cap \mathbb{Z}$, $p_k \geq p_{k+1}, s_k \geq s_{k+1}$, and thus:

$$\forall\ (\boldsymbol{x},\boldsymbol{y}) \in \{(\boldsymbol{x},\boldsymbol{y}) | \|\boldsymbol{x}-\boldsymbol{y}\|_\infty \geq p_{k'}\},\ \mathcal{P}(\boldsymbol{x}; p_k) \neq \mathcal{P}(\boldsymbol{y}; p_k), k \geq k' \tag{3}$$

because the DDF predicted by k^{th} RA module has the same resolution as feature map:

$$\begin{cases} \phi_k[\boldsymbol{x}] \equiv \phi_k[\boldsymbol{y}] \text{ if } \mathcal{P}(\boldsymbol{x}; p_k) = \mathcal{P}(\boldsymbol{y}; p_k) \\ \phi_k[\boldsymbol{x}] \not\equiv \phi_k[\boldsymbol{y}] \text{ if } \mathcal{P}(\boldsymbol{x}; p_k) \neq \mathcal{P}(\boldsymbol{y}; p_k) \end{cases} \tag{4}$$

so that:

$$\begin{cases} \exists\ (\boldsymbol{x},\boldsymbol{y}) \in \{(\boldsymbol{x},\boldsymbol{y}) | \|\boldsymbol{x}-\boldsymbol{y}\|_\infty < p_k\}, & \phi_k[\boldsymbol{x}] \equiv \phi_k[\boldsymbol{y}] \\ \forall\ (\boldsymbol{x},\boldsymbol{y},k) \in \{(\boldsymbol{x},\boldsymbol{y},k) | \|\boldsymbol{x}-\boldsymbol{y}\|_\infty \geq p_{k'}, k \geq k'\}, & \phi_k[\boldsymbol{x}] \not\equiv \phi_k[\boldsymbol{y}] \end{cases} \tag{5}$$

Table 1. Notation of symbols in the manuscript and the supplementary material part.

Symbol	Description
$*$	convolution
$\circ$	composition
$\odot$	element-wise product
$\otimes$	tensor product
$\|$ or $[\cdot,\cdot]$	tensor concatenate at feature channel dimension
$\mathcal{D}$	dissimilarity metric between images
$\mathcal{L}$	loss function
$\mathcal{S}$	smoothness regularization
$\mathcal{P}$	pooling mapping from the original full-resolution image's coordinate
$\mathcal{R}$	a trainable network mapping from images/feature maps to a DDF
$\mathcal{A}$	a trainable network mapping from multiple DDFs to a refined DDF
$\mathcal{W}$	warping function from one image to another via a given DDF
$\mathcal{C}$	one/multi-layer convolution subnetwork
$\boldsymbol{I}^{\mathrm{s}}, \boldsymbol{I}^{\mathrm{t}}$	source & target images
$\boldsymbol{F}^{\mathrm{s}}, \boldsymbol{F}^{\mathrm{t}}$	feature maps extracted from source & target images
ϕ	dense displacement field (DDF)
φ	residual dense displacement field
θ	attribute map with contextual and confidence information
ϑ	incremental attribute map
$\boldsymbol{x}, \boldsymbol{y}$	coordinates at image/DDF

The k^{th} predicted DDF ϕ_k can be decomposited as:

$$\phi_k[\boldsymbol{x}] = \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{x}] + \phi_{k-1}[\boldsymbol{x} - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{x}]] \tag{6}$$

where $\phi_k \circ \phi_{k-1}^{-1}$ is regressed by $\mathcal{R}_k$ and $\mathcal{T}_k$ in RA module as described in the manuscript. The difference between two displacements $\phi_k[x]$ and $\phi_k[y]$ can be written as:

$$\begin{aligned}
&\Delta\phi_k(\boldsymbol{x},\boldsymbol{y}) \\
&:= \phi_k[\boldsymbol{x}] - \phi_k[\boldsymbol{y}] \\
&= \phi_k \circ \phi_{k-1}^{-1} \circ \phi_{k-1}[\boldsymbol{x}] - \phi_k \circ \phi_{k-1}^{-1} \circ \phi_{k-1}[\boldsymbol{y}] \\
&= (\phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{x}] + \phi_{k-1}[\boldsymbol{x} - (\phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{x}])) - (\phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{y}] + \phi_{k-1}[\boldsymbol{y} - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{y}]]) \\
&= \underbrace{\phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{x}] - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{y}]}_{\text{i}} + \underbrace{\phi_{k-1}[\boldsymbol{x} - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{x}]] - \phi_{k-1}[\boldsymbol{y} - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{y}]]}_{\text{ii}}
\end{aligned} \tag{7}$$

where the range of Eq. (7)(i) is limited by the k^{th} RA module and Eq. (7)(ii) can be substituted with $\phi_{k-1}[\boldsymbol{x}'] - \phi_{k-1}[\boldsymbol{y}']$ by $\boldsymbol{x}' := \boldsymbol{x} - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{x}], \boldsymbol{y}' := \boldsymbol{y} - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{y}]$, where the iterative equation of Eq. (7) can be thus written as:

$$\begin{cases}
\Delta\phi_k(\boldsymbol{x}^k,\boldsymbol{y}^k) - (\boldsymbol{x}^k - \boldsymbol{y}^k) = \Delta\phi_{k-1}(\boldsymbol{x}^{k-1},\boldsymbol{y}^{k-1}) - (\boldsymbol{x}^{k-1} - \boldsymbol{y}^{k-1}) \\
\boldsymbol{x}^k - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{x}^k] = \boldsymbol{x}^{k-1} \\
\boldsymbol{y}^k - \phi_k \circ \phi_{k-1}^{-1}[\boldsymbol{y}^k] = \boldsymbol{y}^{k-1}
\end{cases} \tag{8}$$

starting from $\Delta\phi_0(\boldsymbol{x}^0,\boldsymbol{y}^0) := 0\ \forall \boldsymbol{x}^0,\boldsymbol{y}^0 \in \mathbb{Z}^d$. The analytic equation is derived as:

$$\begin{cases} \Delta\phi_k(\boldsymbol{x}^k,\boldsymbol{y}^k) = \Delta\phi_{k''-1}(\boldsymbol{x}^{k''-1},\boldsymbol{y}^{k''-1}) + \sum_{k'=k''}^{k} \Delta\psi_{k'}(\boldsymbol{x}^{k'},\boldsymbol{y}^{k'}) \\ \boldsymbol{x}^k - \boldsymbol{y}^k = (\boldsymbol{x}^{k''-1} - \boldsymbol{y}^{k''-1}) + \sum_{k'=k''}^{k} \Delta\psi_{k'}(\boldsymbol{x}^{k'},\boldsymbol{y}^{k'}) \\ \Delta\psi_{k'}(\boldsymbol{x}^{k'},\boldsymbol{y}^{k'}) := \phi_{k'} \circ \phi_{k'-1}^{-1}[\boldsymbol{x}^{k'}] - \phi_{k'} \circ \phi_{k'-1}^{-1}[\boldsymbol{y}^{k'}] \end{cases} \tag{9}$$

Substitute Eq. (5) into Eq. (9), we can conclude that

$$\begin{aligned} &\exists\ (\boldsymbol{x}^k,\boldsymbol{y}^k) \in \{(\boldsymbol{x},\boldsymbol{y}) | \|\boldsymbol{x}-\boldsymbol{y}\|_\infty < p_{k''-1} + \textstyle\sum_{k'=k''}^{k} \Delta\psi_{k'}(\boldsymbol{x}^{k'},\boldsymbol{y}^{k'})\}, \\ &\quad \Delta\phi_k(\boldsymbol{x}^k,\boldsymbol{y}^k) = \textstyle\sum_{k'=k''}^{k} \Delta\psi_{k'}(\boldsymbol{x}^{k'},\boldsymbol{y}^{k'}); \\ &\forall\ (\boldsymbol{x}^k,\boldsymbol{y}^k,k) \in \{(\boldsymbol{x},\boldsymbol{y},k) | \|\boldsymbol{x}-\boldsymbol{y}\|_\infty \geq p_{k''} + \textstyle\sum_{k'=k''+1}^{k} \Delta\psi_{k'}(\boldsymbol{x}^{k'},\boldsymbol{y}^{k'})\}, k \geq k''\}, \\ &\quad \Delta\phi_k(\boldsymbol{x}^k,\boldsymbol{y}^k) \geq \textstyle\sum_{k'=k''}^{k} \Delta\psi_{k'}(\boldsymbol{x}^{k'},\boldsymbol{y}^{k'}); \end{aligned}$$

with satisfying $\sup(\Delta\phi_k(\boldsymbol{x}^k,\boldsymbol{y}^k)) = \sup(\|\phi_k[\boldsymbol{x}] - \phi_k[\boldsymbol{y}]\|_\infty)$, $\sup(\Delta\psi_k(\boldsymbol{x}^k,\boldsymbol{y}^k)) = 2a_k$, which thus prove Theorem 1 (**Regional Dependency**) in the manuscript.

1.2 Confidence-Weighted Interpolation

The areas lacking texture or structural features usually results in the deviation on prediction and thus requires correction from the interpolation or smoothing based on the neighbouring predicted values. To strength the different displacement at each pixel/voxel with individual weights, the confidence values are respectively quantified by $\mathcal{C}^1$ for φ_k and ϕ_{k-1}.

For example of a simple Gaussian-based smoothing on a single-head residual DDF φ_k adaptively weighted by a confidence map $\boldsymbol{C}$:

$$\begin{aligned} \mathrm{smooth}(\varphi_k, \boldsymbol{C}) &= \overbrace{\boldsymbol{C} \odot (\varphi_k * \mathbf{G})}^{\text{smoothed DDF}} + \overbrace{(1-\boldsymbol{C}) \odot \varphi_k}^{\text{original DDF}} \\ &= \varphi_k - \underbrace{\boldsymbol{C} \odot (\varphi_k * \mathbf{L})}_{=\varphi'_k} \end{aligned} \tag{10}$$

where $\mathbf{G}$ denotes a Gaussian filter kernel for smoothing, $\mathbf{L} := 1 - \mathbf{G}$ denotes the Laplacian filter kernel. Here the Laplacian convolution $(\varphi_k * \mathbf{L})$ is regressed by $\mathcal{C}^2(\phi_k)$, and the confidence weight $\boldsymbol{C} := \mathcal{C}^1(\vartheta_k)$ is implicitly regressed from ϑ_k with general representation for the aim of higher accuracy. Thus the calculation of $\mathrm{smooth}(\phi_{k-1}, \mathcal{C}^1(\theta_{k-1}))$ and $\mathrm{smooth}(\varphi_k, \mathcal{C}^1(\vartheta_k))$ could be regressed in $\mathcal{C}^4([\varphi'_k, \varphi_k, \phi'_{k-1}, \phi_{k-1}])$,

1.3 Multi-Head Disentanglement

To disentangle the predicted DDF with preserving discontinuities and the trend of motions, the Multi-Head masks $\boldsymbol{M} := \mathrm{softmax}(\theta_k)$ is inserted into Eq. (10) for decoupling and smoothing the prediction on the different regions of DDF ϕ_k:

$$\mathrm{smooth}(\phi_{k-1}, \boldsymbol{C}, \boldsymbol{M}) = \phi_{k-1} - \sum_{\{m\}} \underbrace{\boldsymbol{C} \odot ((\boldsymbol{M} \otimes \phi_{k-1}) * \mathbf{L})}_{=\phi'_{k-1}} \tag{11}$$

and M-H residual DDF φ_k:

$$\text{smooth}(\varphi_k, \boldsymbol{C}, \boldsymbol{M}) = \sum_{\{m\}} \varphi_k - \underbrace{\boldsymbol{C} \odot ((\boldsymbol{M} \odot \varphi_k) * \mathbf{L})}_{=\varphi'_k} \tag{12}$$

where $\sum_{\{m\}}$ denotes the head-dimension sum. The calculation of Eq. (11) and Eq. (12) could be regressed in:

$$\phi_k = \mathcal{C}^4([\varphi'_k, \sum_{\{m\}}(\varphi_k), \phi'_{k-1}, \phi_{k-1}]) \tag{13}$$

to predict the ouput DDF of the k^{th} RA module ϕ_k.

2 Network Architecture

The network structure details of encoder, decoder and RA modules are respectively illustrated in Tab. 2, Tab. 3 and Tab. 4.

Table 2. Network of encoder.

layer(s)	kernel	dilation	channels	scale	in	out
conv,norm,act	3	1	1/8	1	$\boldsymbol{I}^{\text{s,t}}$	r1
conv,norm,act	3	1	8/8	1	r1	f1
conv,norm	3	3	8/8	1	f1	f1
act	-	-	8/8	1	f1+r1	s1
downsample	-	-	-	-	s1	s1
conv,norm,act	3	1	8/16	2	s1	r2
conv,norm,act	3	1	16/16	2	r2	f2
conv,norm	3	3	16/16	2	f2	f2
act	-	-	16/16	2	f2+r2	s2
downsample	-	-	-	-	s2	s2
conv,norm,act	3	1	16/16	4	s2	r3
conv,norm,act	3	1	16/16	4	r3	f3
conv,norm	3	3	16/16	4	f3	f3
act	-	-	16/16	4	f3+r3	s3
downsample	-	-	-	-	s3	s3
conv,norm,act	3	1	16/32	8	s3	r4
conv,norm,act	3	1	32/32	8	r4	f4
conv,norm	3	3	32/32	8	f4	f4
act	-	-	32/32	8	f4+r4	s4
downsample	-	-	-	-	s4	s4

Table 3. Network structure of decoder for Residual Aligner Network (RAn_0,RAn_3,RAn_4,RAn_4^+) with varying channels (c_0=32,32,36,48; c_1=64,48,44,48, c_2=48,48,44,48, c_3=32,32,28,32, c_4=24,32,28,32), pooling scales and layer inputs.

layer(s)	ker	dila	chns	RAn_0 scale	RAn_0 in	RAn_3 scale	RAn_3 in	RAn_4 scale	RAn_4 in	RAn_4^+ scale	RAn_4^+ in	out
upsample					×		✓		✓		✓	
conv,norm,act	3	1	c_0/32	16	s4	2	s4,s3	1	s4,s3,s2	1	s4,s3,s2	r5
conv,norm,act	3	1	32/32	16	r5	2	r5	1	r5	1	r5	f5
conv,norm	3	3	32/32	16	f5	2	f5	1	f5	1	f5	f5
act	-	-	32/32	16	f5+r5	2	f5+r5	1	f5+r5	1	f5+r5	$\boldsymbol{F}_0^{s/t}$
upsample					✓		×		×		×	
conv,norm,act	3	1	c_1/32	8	$\boldsymbol{F}_0^{s/t}$|s4	2	$\boldsymbol{F}_0^{s/t}$|s4	1	$\boldsymbol{F}_0^{s/t}$|s4	1	$\boldsymbol{F}_0^{s/t}$|s4	r6
conv,norm,act	3	1	32/32	8	r6	2	r6	1	r6	1	r6	f6
conv,norm	3	3	32/32	8	f6	2	f6	1	f6	1	f6	f6
act	-	-	32/32	8	f6+r6	2	f6+r6	1	f6+r6	1	f6+r6	$\boldsymbol{F}_1^{s/t}$
upsample					✓		×		×		×	
conv,norm,act	3	1	c_2/16	4	$\boldsymbol{F}_1^{s/t}$|s3	2	$\boldsymbol{F}_1^{s/t}$|s3	1	$\boldsymbol{F}_1^{s/t}$|s3	1	$\boldsymbol{F}_1^{s/t}$|s3	r7
conv,norm,act	3	1	16/16	4	r7	2	r7	1	r7	1	r7	f7
conv,norm	3	3	16/16	4	f7	2	f7	1	f7	1	f7	f7
act	-	-	16/16	4	f7+r7	2	f7+r7	1	f7+r7	1	f7+r7	$\boldsymbol{F}_2^{s/t}$
upsample					✓		×		×		×	
conv,norm,act	3	1	c_3/16	2	$\boldsymbol{F}_2^{s/t}$|s2	2	$\boldsymbol{F}_2^{s/t}$|s2	1	$\boldsymbol{F}_2^{s/t}$|s2	1	$\boldsymbol{F}_2^{s/t}$|s2	r8
conv,norm,act	3	1	16/16	2	r8	2	r8	1	r8	1	r8	f8
conv,norm	3	3	16/16	2	f8	2	f8	1	f8	1	f8	f8
act	-	-	16/16	2	f8+r8	2	f8+r8	1	f8+r8	1	f8+r8	$\boldsymbol{F}_3^{s/t}$
upsample					✓		✓		×		×	
conv,norm,act	3	1	c_4/8	1	$\boldsymbol{F}_3^{s/t}$|s1	1	$\boldsymbol{F}_3^{s/t}$|s1	1	$\boldsymbol{F}_3^{s/t}$|s1	1	$\boldsymbol{F}_3^{s/t}$|s1	r9
conv,norm,act	3	1	8/8	1	r9	1	r9	1	r9	1	r9	f9
conv,norm	3	3	8/8	1	f9	1	f9	1	f9	1	f9	f9
act	-	-	8/8	1	f9+r9	1	f9+r9	1	f9+r9	1	f9+r9	s9
conv	1	1	8/d	1	s9	1	s9	1	s9	1	s9	$\boldsymbol{F}_4^{s/t}$

3 Training Detail

3.1 Synthetic Training

For the experiments on inter-subject alignment of abdomen and lung CT, the models are first pre-trained for 100k iteration on synthetic DDF $\tilde{\phi}$ combining rigid spatial transformation with rotation angle $\beta \sim \mathcal{U}(-\pi/4, \pi/4)$ at an arbitrary axis and deformation synthesized by thin plate spline as well as gaussian deformation by 20 random seeds located uniformly randomized within the image domain. , with the loss function set as:

$$\mathcal{L}_{\text{syn}} = \sum \|\phi - \tilde{\phi}\|_2^2 + \lambda \sum \|\nabla \phi\|_2^2 \tag{14}$$

where λ denotes the weight of regularization.

3.2 Real-data Training

Inter-subject Registration Then the inter-subject registration models are trained on real data for 100k iterations with the loss function:

$$\mathcal{L} = \mathcal{D}(\boldsymbol{I}^{\mathrm{t}} - \phi(\boldsymbol{I}^{\mathrm{s}})) + \lambda \|\nabla\phi \odot \mathrm{e}^{-\|\nabla \boldsymbol{I}^{\mathrm{t}}\|_2^2}\|_2^2 \tag{15}$$

where normalized cross correlation and mean squared error are used in abdomen and lung CT respectively for $\mathcal{D}$ following [1].

Intra-subject Registration (Lung) The loss function for training of intra-subject registration models includes one more landmark error term than Eq. (16):

$$\mathcal{L} = \mathcal{D}(\boldsymbol{I}^{\mathrm{t}} - \phi(\boldsymbol{I}^{\mathrm{s}})) + \lambda \|\nabla\phi \odot \mathrm{e}^{-\|\nabla \boldsymbol{I}^{\mathrm{t}}\|_2^2}\|_2^2 + \beta \underbrace{\frac{1}{|\boldsymbol{X}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{X}} \|\boldsymbol{x} - \phi(\boldsymbol{y})\|_2^2}_{\text{landmark error}} \tag{16}$$

where $\boldsymbol{X}$ denotes the set of the corresponding landmarks' coordinates from the pairs of lung CT scans.

4 Additional Results

We compared Residual Aligner Network with the relevant state-of-the-art network structures. The Voxelmorph [1] (VM1/VM2: light-/heavy-weight model) is adopted as the representative method of direct regression (DR). The composite network combing CNN (Cn: Global-net) and U-net (Un: Local-net) following to [2], as well as 5-Recursive Cascaded Network [5] (RCn1/RCn2: light-/heavy-weight model) are also adopted into the framework as the relevant baselines representing multi-stage (MS) networks. Dual-stream Pyramidal network (DPRn) [3] is selected as the baseline for feature pyramidal (FP) networks. Additionally, we also replace RA-module with cross attention (Attn) [4] to compare the performance at module-level.

To clearly show the performance detail of the previous relevant models compared with our RANs as well as the ablation studies on the nine organs: spleen (Fig. 1), right kidney (Fig. 2), left kidney (Fig. 3), esophagus (Fig. 4), liver (Fig. 5), aorta (Fig. 6), inferior vena cava (Fig. 7), portal splenic vein (Fig. 8), and pancreas (Fig. 9).

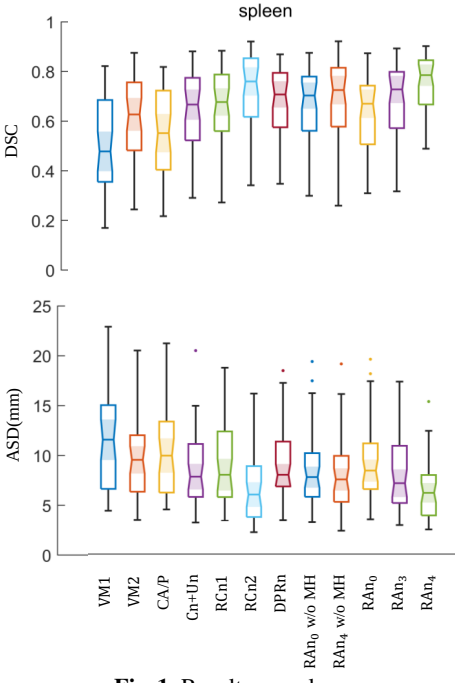

Fig. 1. Results on spleen.

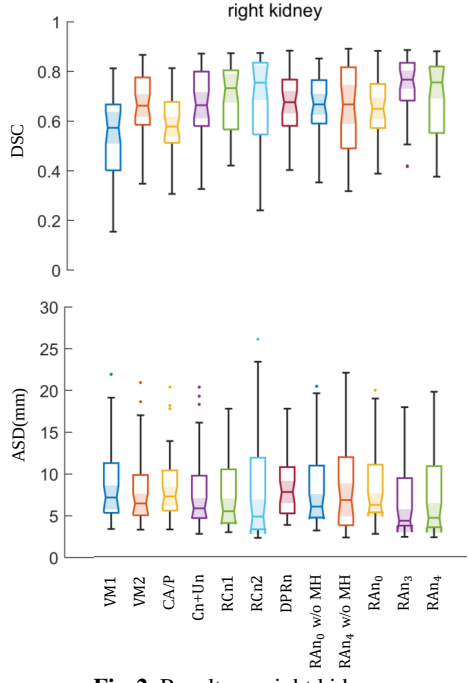

Fig. 2. Results on right kidney.

Table 4. Network structure of Residual Aligner (RA) module for Residual Aligner Network (RAn_0,RAn_3,RAn_4,RAn_4^+) with varying scales and dilation rates.

layer(s)	ker	chns	RAn_0		RAn_3		RAn_4		RAn_4^+		in	out
			scale	dila	scale	dila	scale	dila	scale	dila		
conv,act,conv,act	3	64/18/18	16	1	2	8	1	16	1	16	$\boldsymbol{F}_0^{\text{s}}\|\boldsymbol{F}_0^{\text{t}}$	m0
conv,act,conv	3	18/27/md	16	1	2	1	1	1	1	1	m0	φ_0
conv,act,conv	3	18/18/m	16	1	2	1	1	1	1	1	m0	ϑ_0
conv,norm,act	3	18/1	16	1	2	1	1	1	1	1	ϑ_0	ϑ_0'
conv	1	m/m	16	0	2	0	1	0	1	0	ϑ_0	θ_0
reshape,conv	3	md/m/9	16	1	2	1	1	1	1	1	$\sigma(\theta_0)\odot\varphi_0$	df0
conv,reshape	3	9/1/d	16	1	2	1	1	1	1	1	$\vartheta_0'\odot$df0	ϕ_0
upsample			✓		×		×		×		ϕ_0,θ_0	ϕ_0,θ_0
conv,act,conv,act	3	64/18/18	8	1	2	4	1	8	1	8	$\phi_0(\boldsymbol{F}_1^{\text{s}})\|\boldsymbol{F}_1^{\text{t}}$	m1
conv,act,conv	3	18/27/md	8	1	2	1	1	1	1	1	m1	φ_1
conv,act,conv	3	18/18/m	8	1	2	1	1	1	1	1	m1	ϑ_1
conv,norm,act	3	18/1	8	1	2	1	1	1	1	1	ϑ_1,θ_0	ϑ_1',θ_0'
conv	1	$2m/m$	8	0	2	0	1	0	1	0	$\vartheta_1\|\theta_0$	θ_1
reshape,conv	3	md/m/9	8	1	2	1	1	1	1	1	$\sigma(\theta_1)\odot\varphi_1$	df1
reshape,conv	3	md/m/9	8	1	2	1	1	1	1	1	$\sigma(\theta_1)\otimes\phi_0$	dp1
conv,reshape	3	18/1/d	8	1	2	1	1	1	1	1	$\vartheta_1'\odot$df1$\|\theta_0'\odot$dp1	ϕ_1
upsample			✓		×		×		×		ϕ_1,θ_1	ϕ_1,θ_1
conv,act,conv,act	3	32/18/18	4	1	2	2	1	4	1	4	$\phi_1(\boldsymbol{F}_2^{\text{s}})\|\boldsymbol{F}_2^{\text{t}}$	m2
conv,act,conv	3	18/27/md	4	1	2	1	1	1	1	1	m2	φ_2
conv,act,conv	3	18/18/m	4	1	2	1	1	1	1	1	m2	ϑ_2
conv,norm,act	3	18/1	4	1	2	1	1	1	1	1	ϑ_2,θ_1	ϑ_2',θ_1'
conv	1	$2m/m$	4	0	2	0	1	0	1	0	$\vartheta_2\|\theta_1$	θ_2
reshape,conv	3	md/m/9	4	1	2	1	1	1	1	1	$\sigma(\theta_2)\odot\varphi_2$	df2
reshape,conv	3	md/m/9	4	1	2	1	1	1	1	1	$\sigma(\theta_2)\otimes\phi_1$	dp2
conv,reshape	3	18/1/d	4	1	2	1	1	1	1	1	$\vartheta_2'\odot$df2$\|\theta_1'\odot$dp2	ϕ_2
upsample			✓		×		×		×		ϕ_2,θ_2	ϕ_2,θ_2
conv,act,conv,act	3	32/18/18	2	1	2	1	1	2	1	2	$\phi_2(\boldsymbol{F}_3^{\text{s}})\|\boldsymbol{F}_3^{\text{t}}$	m3
conv,act,conv	3	18/27/md	2	1	2	1	1	1	1	1	m3	φ_3
conv,act,conv	3	18/18/m	2	1	2	1	1	1	1	1	m3	ϑ_3
conv,norm,act	3	18/1	2	1	2	1	1	1	1	1	ϑ_3,θ_2	ϑ_3',θ_2'
conv	1	$2m/m$	2	0	2	0	1	0	1	0	$\vartheta_3\|\theta_2$	θ_3
reshape,conv	3	md/m/9	2	1	2	1	1	1	1	1	$\sigma(\theta_3)\odot\varphi_3$	df3
reshape,conv	3	md/m/9	2	1	2	1	1	1	1	1	$\sigma(\theta_3)\otimes\phi_2$	dp3
conv,reshape	3	18/1/d	2	1	2	1	1	1	1	1	$\vartheta_3'\odot$df3$\|\theta_2'\odot$dp3	ϕ_3
upsample			✓		✓		×		×		ϕ_3,θ_3	ϕ_3,θ_3
conv,act,conv,act	3	16/18/18	1	1	1	1	1	1	1	1	$\phi_3(\boldsymbol{F}_4^{\text{s}})\|\boldsymbol{F}_4^{\text{t}}$	m4
conv,act,conv	3	18/27/md	1	1	1	1	1	1	1	1	m4	φ_4
conv,act,conv	3	18/18/m	1	1	1	1	1	1	1	1	m4	ϑ_4
conv,norm,act	3	18/1	1	1	1	1	1	1	1	1	ϑ_4,θ_3	ϑ_4',θ_3'
conv	1	$2m/m$	1	0	1	0	1	0	1	0	$\vartheta_4\|\theta_3$	θ_4
reshape,conv	3	md/m/9	1	1	1	1	1	1	1	1	$\sigma(\theta_4)\odot\varphi_4$	df4
reshape,conv	3	md/m/9	1	1	1	1	1	1	1	1	$\sigma(\theta_4)\otimes\phi_3$	dp4
conv,reshape	3	18/1/d	1	1	1	1	1	1	1	1	$\vartheta_4'\odot$df4$\|\theta_3'\odot$dp4	ϕ

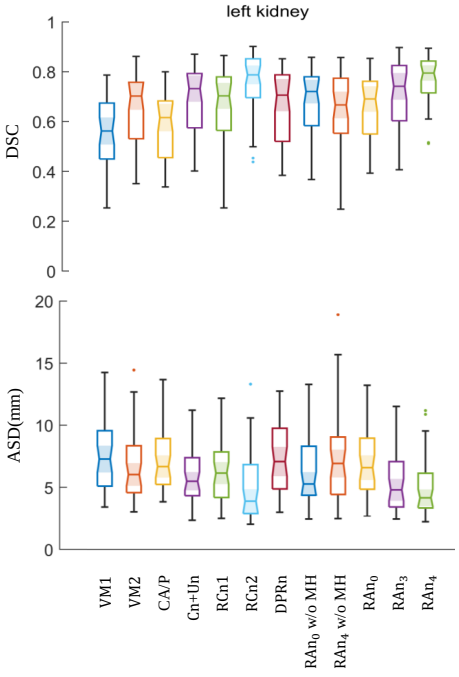

Fig. 3. Results on left kidney.

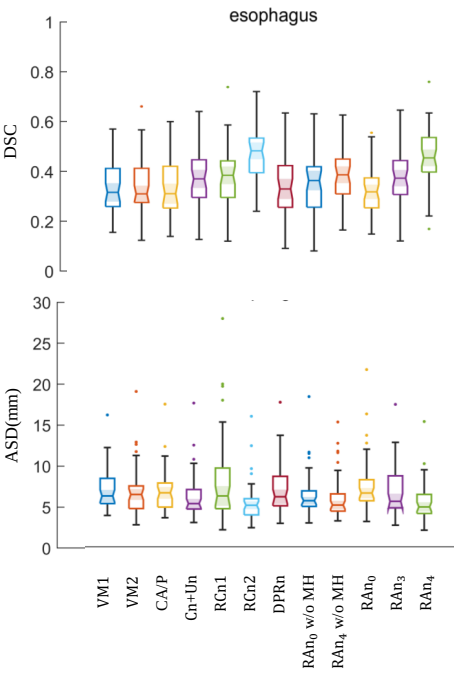

Fig. 4. Results on left kidney.

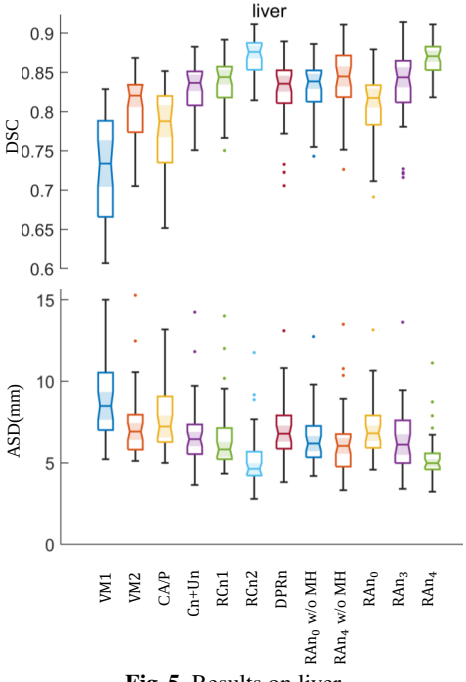

Fig. 5. Results on liver.

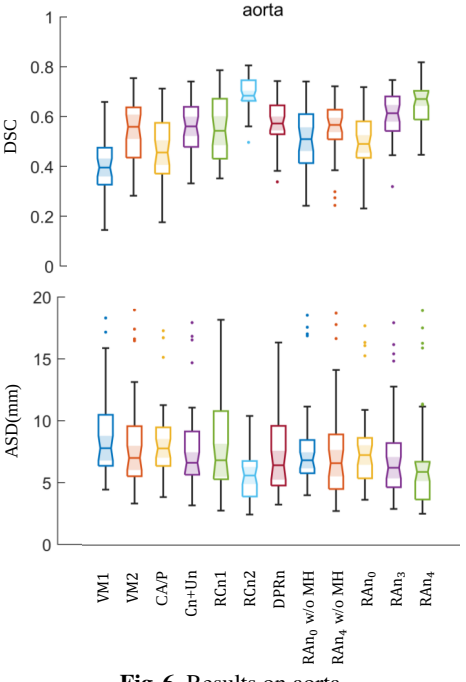

Fig. 6. Results on aorta.

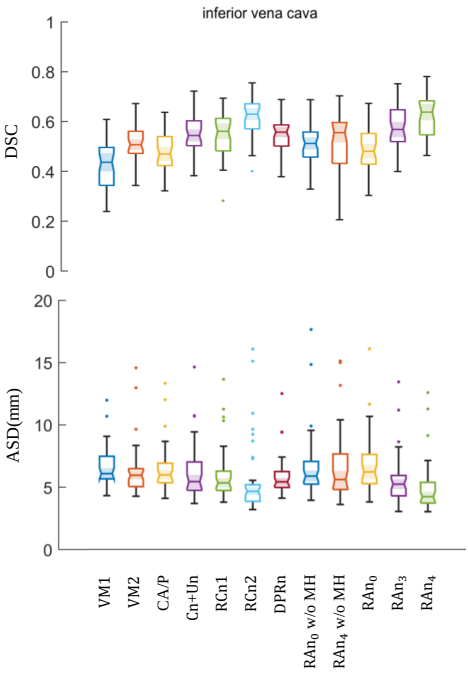

Fig. 7. Results on inferior vena cava.

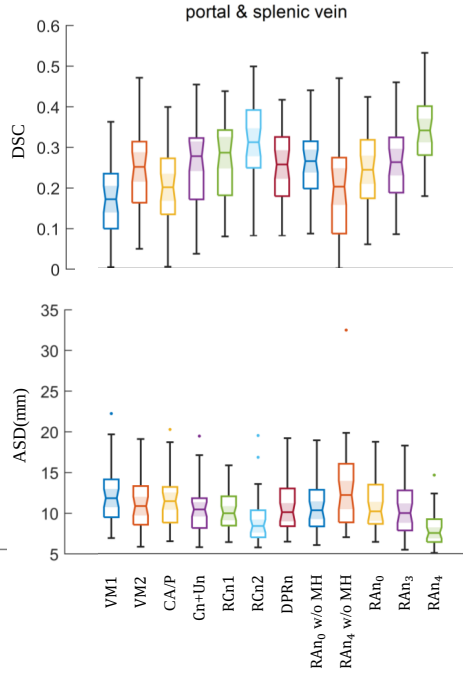

Fig. 8. Results on portal splenic vein.

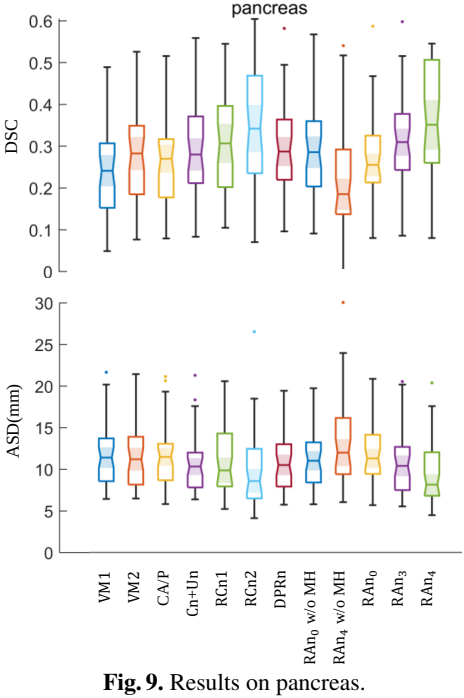

Fig. 9. Results on pancreas.

References

1. Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V.: Voxelmorph: a learning framework for deformable medical image registration. IEEE transactions on medical imaging **38**(8), 1788–1800 (2019) 6
2. Hu, Y., Modat, M., Gibson, E., Li, W., Ghavami, N., Bonmati, E., Wang, G., Bandula, S., Moore, C.M., Emberton, M., et al.: Weakly-supervised convolutional neural networks for multimodal image registration. Medical image analysis **49**, 1–13 (2018) 6
3. Kang, M., Hu, X., Huang, W., Scott, M.R., Reyes, M.: Dual-stream pyramid registration network. Medical Image Analysis **78**, 102379 (2022) 6
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 6
5. Zhao, S., Dong, Y., Chang, E.I., Xu, Y., et al.: Recursive cascaded networks for unsupervised medical image registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10600–10610 (2019) 6