# Dynamic Stream Network for Combinatorial Explosion Problem in Deformable Medical Image Registration

Supplementary Material

## 7. More serious combinatorial explosion problem in registration

Since DMIR needs to model feature relationships between two images, the possible feature combinations are far more numerous than those in a single input task (classification, segmentation), thus leading to a more serious combinatorial explosion problem. To illustrate and rigorously prove this point, we take segmentation (a typical single input task) as an example and compare the complexity of these two tasks through mathematical methods. It is demonstrated that the combinatorial explosion in the registration task is significantly more serious than that in segmentation.

**Complexity of medical image segmentation:** Segmentation requires assigning a label to each pixel. Suppose each pixel has $L$ possible labels, then the total number of possible segmentation configurations is

$$|C| = L^N = L^{HW}. \tag{11}$$

which is exponential growth in the number of pixels $N$.

**Complexity of deformable medical image registration:** DMIR aims to find correspondences between features in two or more images, often requiring pairwise or combination f features. Given two images, each with $N = H \times W$ feature points, the goal is to identify correct matches. For each feature point $f_i$ in the source image, there exists a candidate set of potential matches in the target image. Let the average size of this candidate set be proportional to the image size, modeled as $c = \alpha N$, where $\alpha$ is the average number of possible feature relationships for each pair of feature combinations. Assuming independent choices for each feature, the total number of possible feature combinations is:

$$|\mathcal{H}| = c^N = (\alpha HW)^{HW}. \tag{12}$$

This expresses an exponential growth of the possible feature combinations with respect to $H \times W$.

**Proof of the more serious combinatorial explosion problems in registration:** The total number of possible segmentation configurations is $|C| = L^N = L^{HW}$, where $L$ is the fixed number of labels, and $N = H \times W$ is the number of pixels. The total number of possible feature combinations in registration is $|\mathcal{H}| = c^N = (\alpha N - 1)^N = (\alpha HW - 1)^{HW}$, where $c = \alpha N - 1$ is the average candidate set size per feature point and grows linearly with $N$. To compare their growth rates, consider the logarithms of these quantities:

$$\log |C| = N \log L, \tag{13}$$

and

$$\log |\mathcal{H}| = N \log(\alpha N - 1). \tag{14}$$

The ratio of these logarithms is

$$R = \frac{\log |\mathcal{H}|}{\log |C|} = \frac{N \log(\alpha N - 1)}{N \log L} = \frac{\log(\alpha N - 1)}{\log L}. \tag{15}$$

As $N$ grows large,

$$\log(\alpha N - 1) \approx \log N + \log \alpha \to \infty, \tag{16}$$

while $\log L$ is constant. Therefore,

$$\lim_{N \to \infty} R = \lim_{N \to \infty} \frac{\log(\alpha N - 1)}{\log L} = \infty. \tag{17}$$

This shows that the logarithm of the registration complexity grows faster than that of segmentation complexity, implying

$$|\mathcal{H}| \gg |C| \quad \text{as} \quad N \to \infty. \tag{18}$$

In other words, the combinatorial explosion in registration tasks is more serious than in segmentation tasks.

## 8. Technical Details

### 8.1. Details of registration architecture

**Details of DySNet:** As show in Fig.7-a, DySNet is designed as a versatile and flexible bidirectional registration architecture. It first forms a bidirectional feature modeling component through alternating DSB modules, and then stacks this feature modeling component to obtain the overall network. Components can use different feature processing methods among themselves. This module alternately processes the features from the two input images to model complex feature relationships. Following Xmorpher [33], $\mathcal{L}_{reg}$ is a deformable registration loss composed of smoothness loss $\mathcal{L}_{smo}$ (Jacobian matrix), which preserves deformation topology, and similarity loss $\mathcal{L}_{sim}$ (mean dice similarity coefficients ), which aligns similar regions.

**Details of DySNet-X:** As show in Fig.7-b, DySNet-X is a variant of DySNet. In the original Xmorpher [33] architecture, the CAT block is replaced by the proposed DSB for feature modeling. By replacing the CAT [33] block with DSB, DySNet-X inherits the bidirectional and symmetric modeling advantages of DySNet and performs registration based on the dynamic feature modeling provided by DSB. This leads to improved registration accuracy and more consistent deformation fields, as DSB significantly strengthens

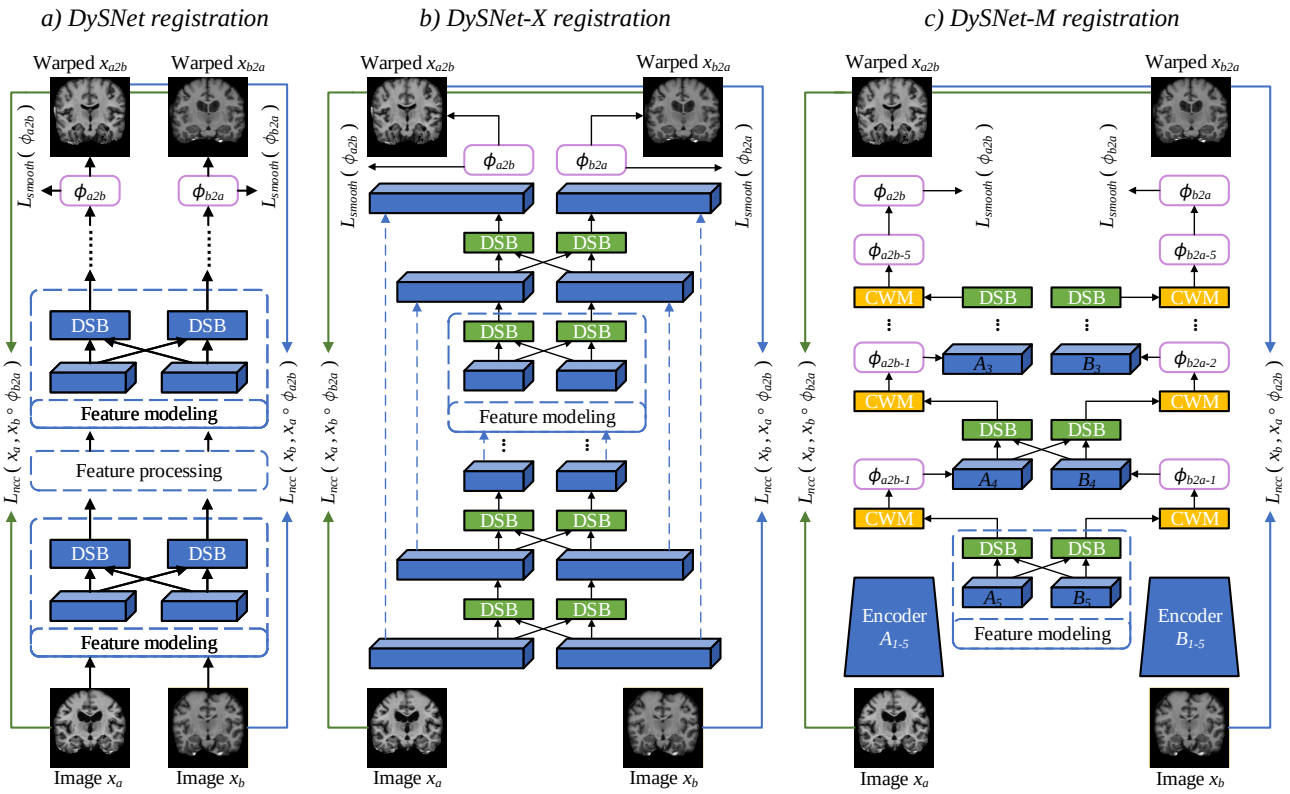


Figure 7. The overall architecture of DySNet and its two instantiated networks are presented. a) DySNet models the input features through alternately stacked DSB modules. Based on this general network, two instantiated networks are derived: b) DySNet-X (based on Xmorpher), c) DySNet-M (based on ModeT).

the symmetry prior and can more effectively perform feature modeling.

**Details of DySNet-M:** As show in Fig.7-c, DySNet-M is achieved by a symmetric pyramid registration structure (ModeT [38]). It uses an encoder with 5 convolutional layers to extract hierarchical features, producing feature maps $A_1 - A_5$ and $B_1 - B_5$. Feature map $A_5$ and $B_5$ go through DSB and CWM [38] to obtain deformation $\phi_{a2b-1}$ of $A_4$. The deformed $A_4$ and $B_4$ input DSB and CWM to generate subfields $\phi_{a2b-2}$. Similar steps apply to $B_3$ and $A_3$. Finally, the $\phi_{a2b}$ obtained from the network warps the image $x_a$ to obtain the registration result, and the processing of image $x_b$ is carried out in the same way.

## 8.2. Details of DySNet's attention mechanisms

Building upon the formulations in Sections 3.1 and 3.2, we further elaborate on the details and properties of the dynamic deformable attention mechanisms in DySNet.

**Multi-head attention projection and reshaping:** Given input features $f^a, f^b \in \mathbb{R}^{B\times C\times H\times W}$, the linear projections for queries, keys, and values are implemented as convolutional layers followed by reshaping:

$$Q = \mathrm{Conv}_q(f^a), \quad K = \mathrm{Conv}_k(f^b), \quad V = \mathrm{Conv}_v(f^b), \tag{19}$$

where each of $Q, K, V \in \mathbb{R}^{B\times C\times H\times W}$ is reshaped to multi-head representation:

$$Q, K, V \to \mathbb{R}^{B\times d\times h\times H\times W}, \tag{20}$$

with $d = \frac{C}{h}$ the per-head channel dimension and $h$ the number of attention heads.

**Interpolation as continuous sampling operator:** The interpolation function $I(\cdot)$ used to sample deformed keys and values at coordinates $D_{N^d}(i)_j \in \mathbb{R}^2$ is implemented as bilinear interpolation, which can be expressed as:

$$I(F, p) = \sum_{q\in\mathcal{N}(p)} w_q(p) F(q), \tag{21}$$

where $\mathcal{N}(p)$ are the four integer grid neighbors of the fractional coordinate $p$, and $w_q(p)$ are bilinear interpolation weights satisfying $\sum_q w_q(p) = 1$. This ensures the interpolation is continuous and differentiable with respect to $p$, enabling backpropagation through the dynamic offset $\Delta i$.

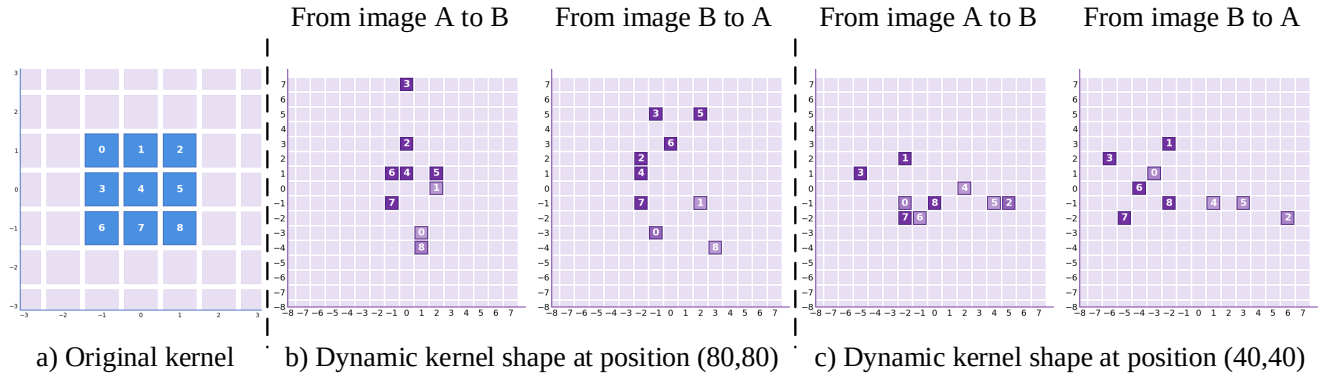


Figure 8. The figure shows the deformation of the convolution kernel. a) Illustrates the original fixed 3x3 convolution kernel. b) and c) are the visualizations of the dynamic kernels at the first layer position (80, 80) and the second layer position (40, 40) of the DySNet-X network, respectively. The shape of the kernel is adaptively adjusted according to the input features.

**Offset prediction network and smoothness:** The offset prediction network $\theta_{\text{offset}}$ is a small CNN module taking concatenated features $X = [f^a, f^b] \in \mathbb{R}^{B\times 2C\times H\times W}$ and predicting offsets:

$$\Delta i = \theta_{\text{offset}}(X) \in \mathbb{R}^{B\times|U_{N^d}|\times d\times H\times W}. \tag{22}$$

The smoothness of $\Delta i$ over spatial positions is implicitly encouraged by the convolutional nature of $\theta_{\text{offset}}$ and regularization losses (if any), leading to locally coherent deformations.

**Joint optimization perspective:** The parameters of $\theta_{\text{offset}}$, $\text{Conv}_q$, $\text{Conv}_k$, $\text{Conv}_v$, and output projection are optimized end-to-end, enabling the network to jointly learn optimal dynamic receptive fields $D_{N^d}(i)$ and corresponding attention weights $\rho^i_{N^d,1}$ that minimize the target loss. This synergy allows DySNet to adaptively focus on spatially relevant features and handle local deformations with precise and smooth spatial attention.

## 9. Experiment Details

### 9.1. Datasets

**MM-WHS [48]** The dataset consists of 120 three-dimensional cardiac image volumes, covering 60 CT scans and 60 MRI scans, all of which were collected from real clinical environments. The images in the dataset are all labeled with seven major anatomical structures of the heart, including the left and right ventricles, left and right atria, left ventricular myocardium, ascending aorta, and pulmonary artery. In this study, the CT modality was used as the training set, and the cardiac region in the images was cropped and resampled to a size of 144×144×128 before being normalized.

**ASOCA [10]** The data focus on the automatic segmentation of coronary arteries, including 60 cases of CT coronary angiography (CTCA), among which 30 cases are normal without any lesions, and 30 cases are patients with different degrees of coronary artery diseases. The dataset is equipped with a complete coronary artery tree structure annotated jointly by multiple experts, covering the entire tree structure of the left and right coronary arteries. The cardiac region in the images was cropped and resampled to a size of 144×144×128 before being normalized.

**CAT08 [15]** The dataset consists of 32 three-dimensional CT images with labels indicating the structure of the heart, covering the annotations of the main chambers of the heart and major blood vessels. The cardiac region in the images was cropped and resampled to a size of 144×144×128 before being normalized.

**PPMI [29]** It is extracted from the PPMI database which is a large Parkinson progression marker initiative database, for 837 T1 brain MR volumes. We resize and crop 160×160×128 volumes on the brain regions, and then normalize the intensity. We also extract the brain regions via HD-BET [17] to avoid the interference of background.

**CANDI [20]** The Child and Adolescent Neuro Development Initiative (CANDI) dataset has 103 T1 brain MR volumes from 57 males and 46 females. Totally 28 brain tissue regions are annotated for masks. We resize and crop 160×160×128 volumes on the brain regions, and then normalize the intensity.

**OASIS-1** This dataset comprises a cross-sectional sample of 416 participants aged between 18 and 96. Each participant has 3 to 4 T1-weighted MRI scans obtained from individual single scans. The brain region was cropped and resampled to a size of 160×160, and the data has already been standardized by the official.

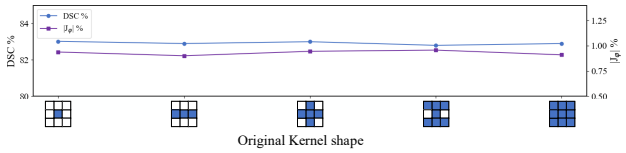


Figure 9. The figure shows the performance of the DySNet network under different shapes of the original kernels and the number of points. The x-axis represents the shape of the original kernel, and the y-axis represents the DSC % and $|J_\phi|\%$. The experimental results demonstrate that, despite the changes in the shape and number of points of the original kernel, DySNet still maintain stable and excellent performance, verifying its ability to dynamically adjust the receptive field and weights.

### 9.2. Implementations

Our DySNet were conducted on NVIDIA RTX 6000 with 24 GB of memory, using PyTorch [31]. We used AdamW [27] as the optimizer with an initial learning rate of $10^{-4}$. The total number of training rounds is 200 with a batch size of 1. To conduct a fair comparison, all the methods employed in our experiment either adopted the same hyperparameter settings or adhered to the experimental settings provided in the original papers.

## 10. More Framework Analysis and Results

### 10.1. Visualization of dynamic kernel

As shown in the Fig.8, the dynamic kernel of DySNet-X adaptively adjust the shape and weights (shade of purple indicates the magnitude of the weight) of the window according to the different input data, achieving more flexible feature modeling. The shape of the dynamic kernel at the first layer of DySNet-X at the position (80, 80) significantly changed with the exchange of the registered images, demonstrating its ability to accurately capture local features. The dynamic kernel at the position (40, 40) of DySNet-X in the second layer showed different deformations, indicating that the network performed adaptive modeling of the structural information of the image at a deeper level. This dynamic adjustment mechanism helps to enhance the model's adaptability to spatial deformation and complex textures, thereby strengthening the overall feature modeling ability.

### 10.2. Analysis of original kernel shape

As shown in the Fig.9, in addition to adjusting the size of the kernel, we also verified the dynamic adaptability of the DySNet network when dealing with different original kernel shapes through this experiment. The results indicated that the performance of DySNet changed very little under different kernel shapes and the number of points, fully demonstrating the dynamic nature and robustness of the network. This not only reduces the reliance on the selection of kernel shapes but also enhances the generalization ability of the model, reflecting the advantage of dynamic in the design concept of DySNet.

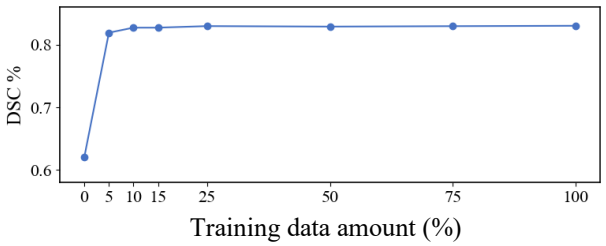


Figure 10. The figure shows the performance changes of DySNet in the 2D brain assessment when using different amounts of training data (%).

### 10.3. Analysis of training data amount

As shown in Fig.10, we evaluate the variation of our DySNet's performance with the enlarging of the training data amount on our 2D brain evaluation. Our DySNet brings a significant improvement even though only 5% training data is involved. When further enlarging the training dataset, the gain of performance gradually decreases owing to the similarity of medical images in the training dataset. Fortunately, our DySNet possesses a powerful ability to model dynamic features and can effectively learn from details. As a result, the performance of this model is still improving gradually.