Equivariant Image Registration

CVPR failed submission.

Presentation Steps:

1: Define equivariance for image registration

2: Select how equivariant we want our algorithm to be

3: Find an architecture that has the equivariance we selected

4: Boost that architecture with a multistep approach, while preserving equivariance

5: Train end to end

Definition of equivariance

A function is equivariant if its input and output are acted on by the same symmetry group, and the function commutes with the action of the group.

F is our function, G is our group,

F equivariant := for all $g \in G$, for all x, $F(g(x)) = g(F(x))$

For example:

Convolution is equivariant to integer translations.

Given a convolution Conv, for any translation U, Conv(X \circ U) = Conv(X) \circ U

(note: translations Act on images by postcomposing with them)

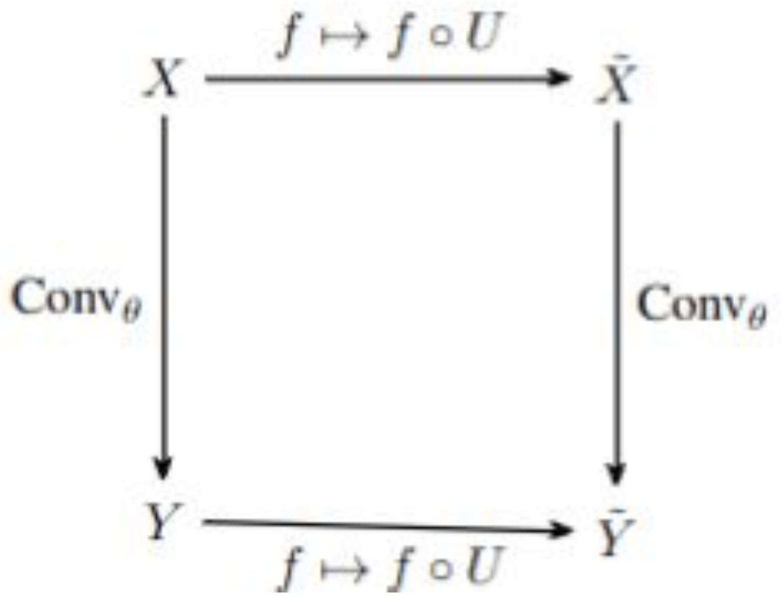


Image Registration Equivariance

To find out how image registration is equivariant, we investigate a setting where registration is analytically tractable: Namely, where our images have the same number of channels as dimensions, and are also diffeomorphisms. Then we solve for an algorithm Xi that perfectly registers them.

$$
\begin{aligned}
I^{\mathrm{M}} \circ \Xi[I^{\mathrm{M}}, I^{\mathrm{F}}] &= I^{\mathrm{F}} \\
(I^{\mathrm{M}})^{-1} \circ I^{\mathrm{M}} \circ \Xi[I^{\mathrm{M}}, I^{\mathrm{F}}] &= (I^{\mathrm{M}})^{-1} I^{\mathrm{F}} \\
\Xi[I^{\mathrm{M}}, I^{\mathrm{F}}] &= (I^{\mathrm{M}})^{-1} I^{\mathrm{F}}
\end{aligned}
$$

Image Registration Equivariance

Ξ is equivariant to the group Diff($\mathbb{R}^N$) × Diff($\mathbb{R}^N$)

$$
\begin{tikzcd}
(I^{\mathrm{M}}, I^{\mathrm{F}}) \arrow[r, "{x, y \mapsto x \circ W, y \circ U}"] \arrow[d, "\Xi"'] & (I^{\mathrm{M}} \circ W, I^{\mathrm{F}} \circ U) \arrow[d, "\Xi"] \\
(I^M)^{-1} \circ I^B \arrow[r, "\varphi \mapsto W^{-1} \circ \varphi \circ U"'] & W^{-1} \circ (I^M)^{-1} \circ I^B \circ U
\end{tikzcd}
$$

We can’t actually demand this full equivariance for arbitrary images.

Assume we have a fully equivariant registration algorithm Ψ

Then register two images that are zero everywhere.

Ψ satisfies $\Psi[A \circ W, B] = W^{-1} \circ \Psi[A, B]$

Then, for any W , $A \circ W = A$ so we need $\Psi[0, 0] = W^{-1} \circ \Psi[0, 0]$, or

$id = W^{-1}$. Contradiction.

Solution: Restrict equivariance

Some possible restrictions

1: Demand that U == W, U and W are translations
This leads to voxelmorph

2: Require that U and W be translations “locally”, but they can be different

This is the regime we are currently working in in this work

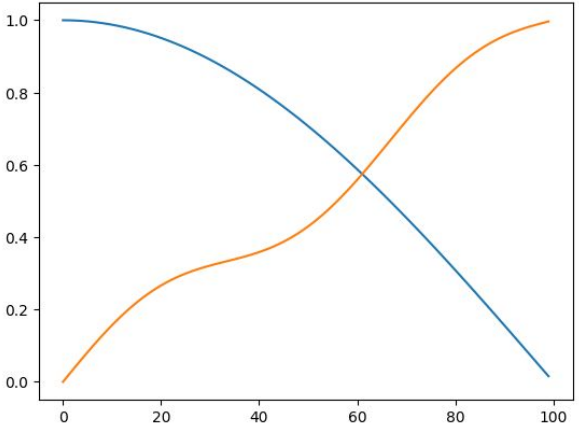
An equivariant architecture

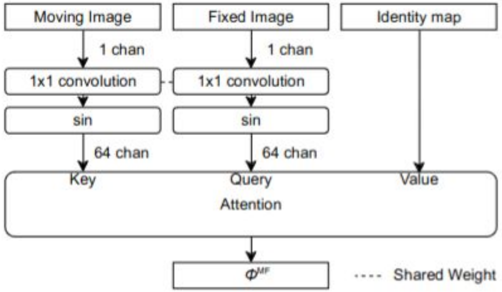
First, we will solve registration for the Diffeomorphism Diffeomorphism problem. The key is that transformers naturally implement function inversion. This is equivariant to arbitrary U, W

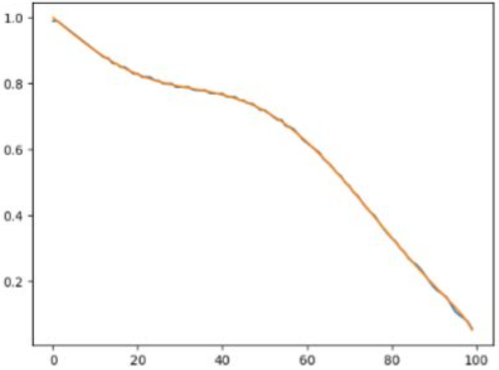
Image A is $A : [0,1] \to [0,1], x \mapsto \cos(\frac{\pi}{2}x)$

Image B is $B : [0,1] \to [0,1], x \mapsto x + 0.07\sin(3\pi x)$

$$\Xi[A,B] = A^{-1} \circ B = \frac{2}{\pi}\cos^{-1}(x + 0.07\sin(3\pi x))$$

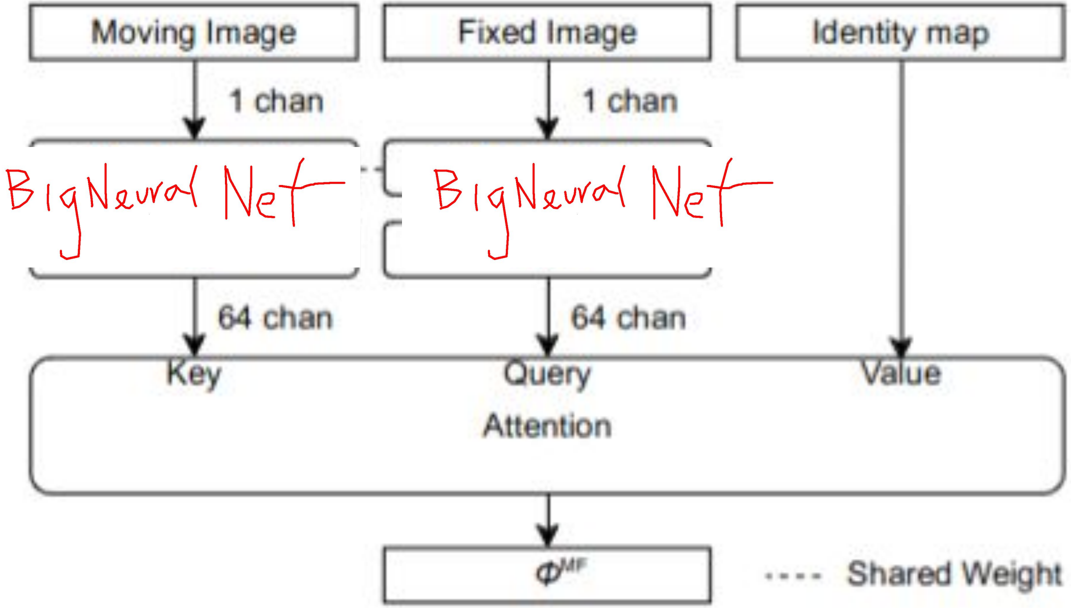






Generalize to Medical Image Registration

Equivariant

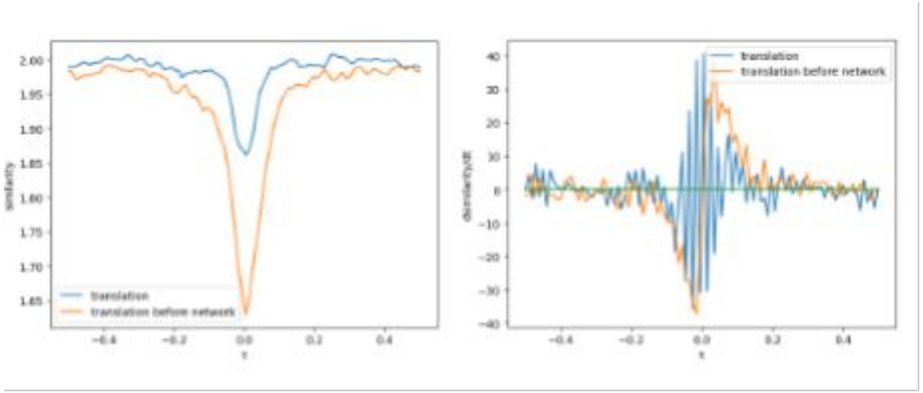


Boost with multistep registration

This is important because training as part of a two step pipeline has better capture radius.

$$\mathcal{T}[I^A, I^B](\vec{r}) = \vec{r} + \begin{bmatrix} t \\ 0 \end{bmatrix} \tag{12}$$

The optimal value of t is zero since there is no bias towards left or right shift of images in this dataset.



Compare loss landscapes of T vs TwoStep{T,Φ}

Standard ConvNet registration via “just predict displacement bro” is translation equivariant restricted to W == U

$$\Phi_\theta[I^{\mathrm{M}}, I^{\mathrm{F}}](\vec{x}) = \mathrm{Conv}_\theta[\mathrm{cat}(I^{\mathrm{M}}, I^{\mathrm{F}})](\vec{x}) + \vec{x}$$

$$\begin{aligned}
\Phi_\theta[I^{\mathrm{M}} \circ U, I^{\mathrm{F}} \circ U](\vec{x}) = & \\
&= \mathrm{Conv}_\theta[\mathrm{cat}(I^{\mathrm{M}} \circ U, I^{\mathrm{F}} \circ U)](\vec{x}) + \vec{x} \\
&= \mathrm{Conv}_\theta[\mathrm{cat}(I^{\mathrm{M}}, I^{\mathrm{F}}) \circ U](\vec{x}) + \vec{x} \\
&= \left(\mathrm{Conv}_\theta[\mathrm{cat}(I^{\mathrm{M}}, I^{\mathrm{F}})] \circ U\right)(\vec{x}) + \vec{x} \\
&= \mathrm{Conv}_\theta[\mathrm{cat}(I^{\mathrm{M}}, I^{\mathrm{F}})](U(\vec{x})) + \vec{x} \\
&= \mathrm{Conv}_\theta[\mathrm{cat}(I^{\mathrm{M}}, I^{\mathrm{F}})](U(\vec{x})) + (\vec{x} + \vec{r}) - \vec{r} \\
&= U^{-1} \circ \left(\mathrm{Conv}_\theta[\mathrm{cat}(I^{\mathrm{M}}, I^{\mathrm{F}})](\vec{x}) + \vec{x}\right) \circ U
\end{aligned}$$

if Φ is equivariant, and Ψ is equivariant only when U == W for U and W from a class of transforms T , then TwoStep{Φ, Ψ} is [W, U] equivariant

$$
\begin{aligned}
&TwoStep\{\Phi, \Psi\}[A \circ W, B \circ U] \qquad (1)\\
&= \Phi[A \circ W, B \circ U] \circ \Psi\left[A \circ W \circ \Phi[A \circ W, B \circ U], B \circ U\right]\\
&= W^{-1} \circ \Phi[A, B] \circ U \circ \Psi\left[A \circ W \circ W^{-1} \circ \Phi[A, B] \circ U, B \circ U\right]\\
&= W^{-1} \circ \Phi[A, B] \circ U \circ \Psi\left[A \circ \Phi[A, B] \circ U, B \circ U\right]\\
&= W^{-1} \circ \Phi[A, B] \circ U \circ U^{-1} \circ \Psi\left[A \circ \Phi[A, B], B\right] \circ U\\
&= W^{-1} \circ \Phi[A, B] \circ \Psi\left[A \circ \Phi[A, B], B\right] \circ U\\
&= W^{-1} \circ TwoStep\{\Phi, \Psi\}[A, B] \circ U
\end{aligned}
$$

So, this architecture is W, U equivariant

$$
\begin{aligned}
&TwoStep\{ \\
&TwoStep\{ \\
&DownsampleRegistration\{Equivariant\_reg\}, \\
&\Psi_1 \\
&\}, \\
&\Psi_2 \\
&\}
\end{aligned}
\tag{1}
$$

where Ψ_i are standard registration U-Nets that are naturally [U, U] equivariant.

Train end to end

To make this train end to end, we made several fine adjustments to equivariant_reg.

1. We normalize the feature vectors before passing to the attention layer, and then scale by 4.

2. We use a simple convolutional network for featurizing with residual connections and no downsampling. We pad the images before processing. For the moving image, we pad the identity map to match, extrapolating. For the fixed image, we just crop back to the original size. This preserves exact translational equivariance, at least for integer shifts.

   We add batch normalization to the feature extraction network.

3. We use diffusion regularization on the end to end registration pipeline.

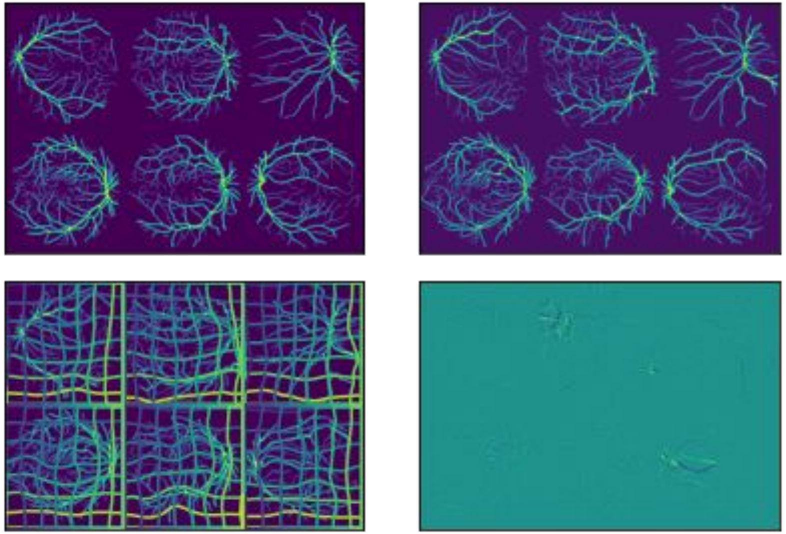
   Life Hack: torch 2.0 attention is O(N) memory instead of O(N^2) if you're careful with stride and channel counts

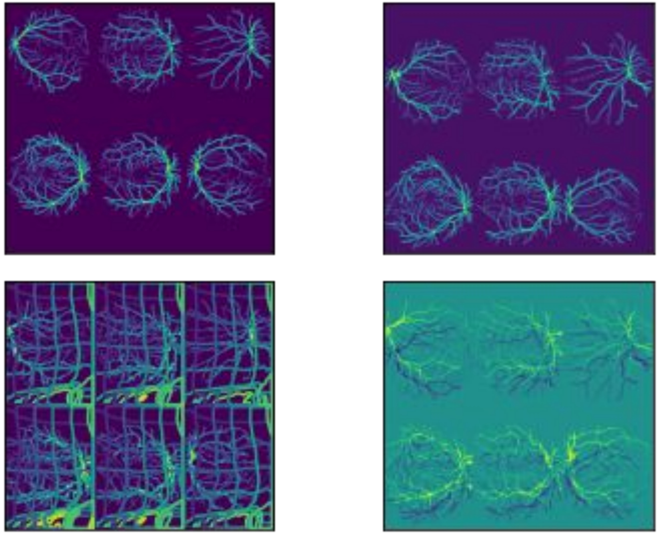
4. We use 128 dimensional feature vectors. 64 dimensions is too little.

Final loss is just Lsim + Lreg, train from scratch.

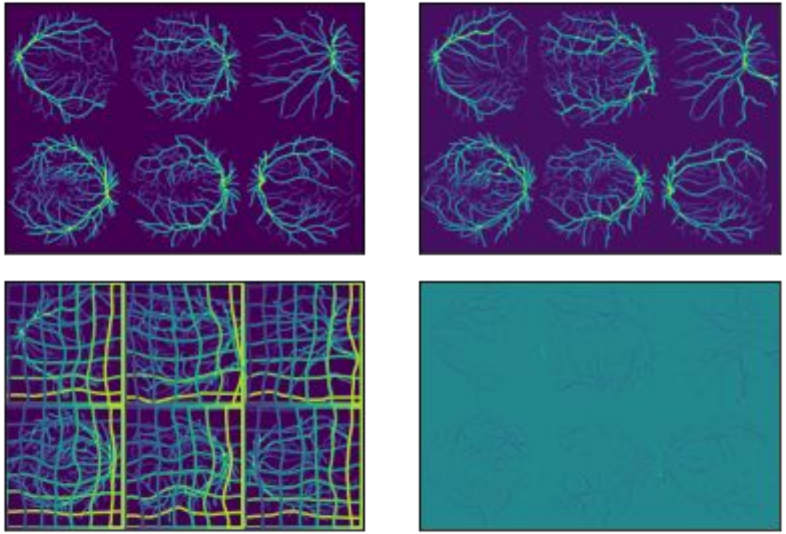
Results:

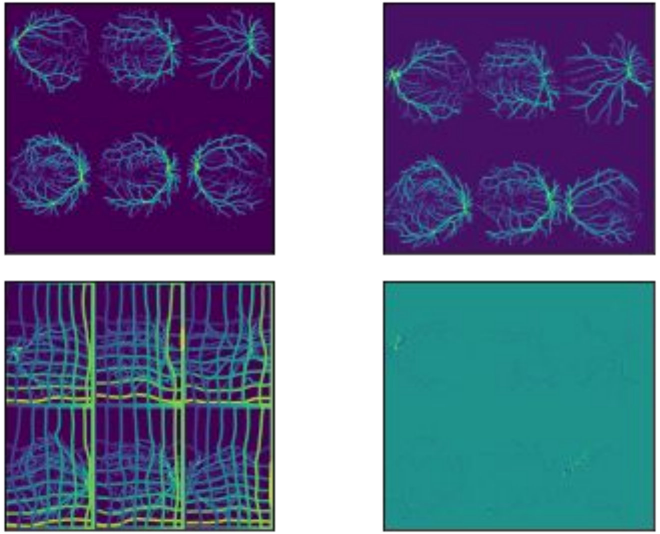
GradICON can register synthetically warped retina segmentations, but not synthetically warped retina segmentations that are also shifted by a constant vector.





Our equivariant architecture can’t tell that the images are shifted, so it solves both equally well.





It can sort of register lung images, but not as well as top algorithms.

Mtre of ~ 4 mm.

mysterious.