Non-iterative Coarse-to-Fine Transformer Network for Joint Segmentation and Registration

JiaXin Xie, Yau-Meng Wong
HKUST COMP 5423

Introduction | Image Registration

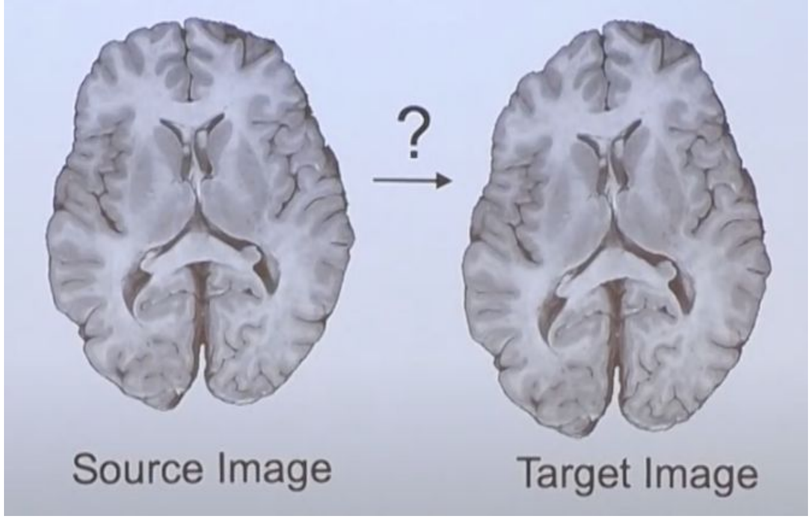
Image Registration Goal:

To spatially align two images from different modalities and/or taken at different times and angles.

diagnosis

disease progression monitoring

treatment planning



cohort studies

image guided intervention

organ atlas creation

Literature Review | Overview of Image Registration Techniques

Registration with Deep Learning

- Key Approach: Coarse-to-fine registration
  Examples: DLIR, mlVIRNET, ULAE-Net, LapIRN
- Key Challenge: How to perform coarse-to-fine registration efficiently within a short runtime?

- Key Approach: Unsupervised
- Key Challenge: Is there a way to introduce supervision to registration task when ground truth is difficult or expensive to obtain?

Literature Review | Overview of Image Registration Techniques

Registration with Deep Learning

- Key Approach: Coarse-to-fine registration
  Examples: DLIR, mlVIRNET, ULAE-Net, LapIRN
- Key Challenge: How to perform coarse-to-fine registration efficiently within a short runtime?

Literature Review | Overview of Image Registration Techniques

Registration with Deep Learning

- Key Approach: Coarse-to-fine registration
  Examples: DLIR, mlVIRNET, ULAE-Net, LapIRN
- Key Challenge: How to perform coarse-to-fine registration efficiently within a short runtime?

→ **NICE-Trans: Non-Iterative Coarse-to-finE Transformer Network**

- Performs joint affine and deformable registration
- Utilizes the skip connections in autoencoder architecture to avoid repeated feature extraction that is necessary in regular coarse-to-find networks
- Leverages Swin transformer to improve model's understanding of the image
  - Can capture more subtle features and connect image information that appears at widely separated positions within input data

Literature Review | Overview of Image Registration Techniques

Registration with Deep Learning

- Key Approach: Unsupervised
- Key Challenge: Is there a way to introduce supervision to a registration model when ground truth is difficult/expensive to obtain?

Literature Review | Overview of Image Registration Techniques

Registration with Deep Learning

- Key Approach: Unsupervised
- Key Challenge: Is there a way to introduce supervision to a registration model when ground truth is difficult/expensive to obtain?

→ **Segmentation-Assisted Registration Networks**

Example: DeepAtlas

- two-two model(two encoders + two decoders)
- one-two model(one encoder + two decoders)

Achieve mutual learning via joint loss

→ **PGCNet: Progressively Coupling Network for Brain MRI Registration**

- “Progressive coupling” design adds feature interaction between segmentation and registration branches

Literature Review | Research Motivation

Research Question 1 Will incorporating a segmentation component to NICE-Trans further improve its registration outcome?

Research Question 2 Is the potential benefit of adding a segmentation branch significant when the model computation cost is taken into consideration?

Key Contributions | NICE-Transeg

Contribution 1 Drawing inspiration from NICE-Trans and PGCNet, we propose NICE-Transeg, a joint 2D image segmentation and registration network. The model modifies NICE-Trans to include **2 segmentation branches** that are "**progressively coupled**" with the registration decoder.

Contribution 2 Our result suggests that NICE-Trans' registration result does not seem to benefit from the design of progressively coupling segmentation and registration branches. **However, the segmentation results of NICE-Transeg suggest that segmentation benefits significantly from the task coupling.**

Contribution 3 We made available a preprocessed, 2D version of the IXI dataset for future researchers with limited computational resources.

NICE-Transeg Model Architecture

$R_\theta (I_m, I_f) = \phi, S_M, S_F$

I_M - single-channel, moving image in 2D spatial domain

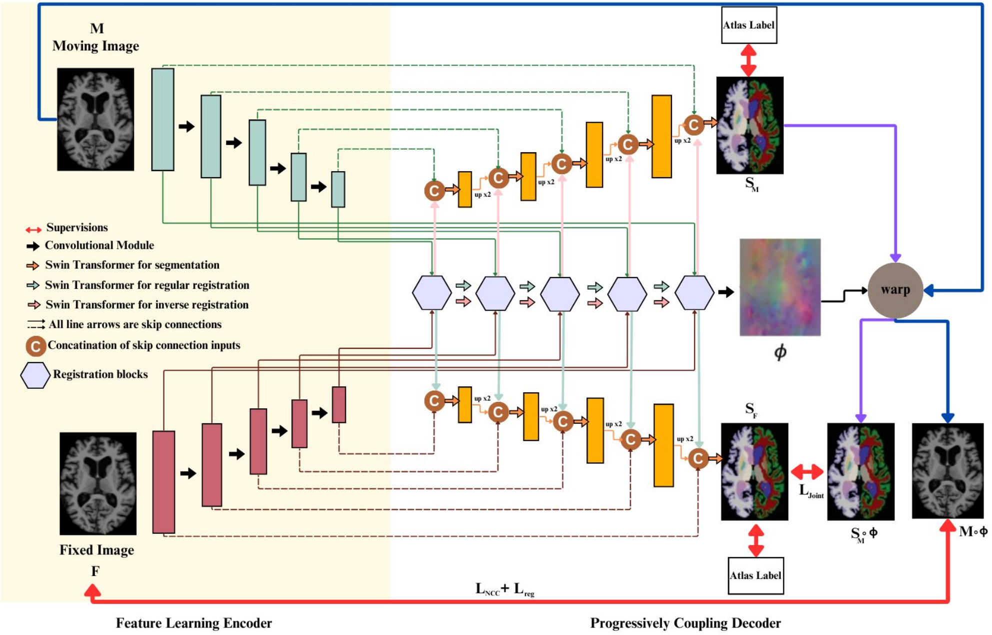
I_F - single-channel, fixed image in 2D spatial domain

θ - Learnable parameters

ϕ - Spatial Transformation that that applies a deformation field to an image

S_M - Segmentation result of moving image

S_F - Segmentation result of fixed image


M
Moving Image
Atlas Label
S_M
up x2
C
Supervisions
Convolutional Module
Swin Transformer for segmentation
Swin Transformer for regular registration
Swin Transformer for inverse registration
All line arrows are skip connections
Concatination of skip connection inputs
Registration blocks
ϕ
warp
S_F
L_{Joint}
$S_M \circ \phi$
$M \circ \phi$
Fixed Image
F
Atlas Label
$L_{NCC} + L_{reg}$
Feature Learning Encoder
Progressively Coupling Decoder

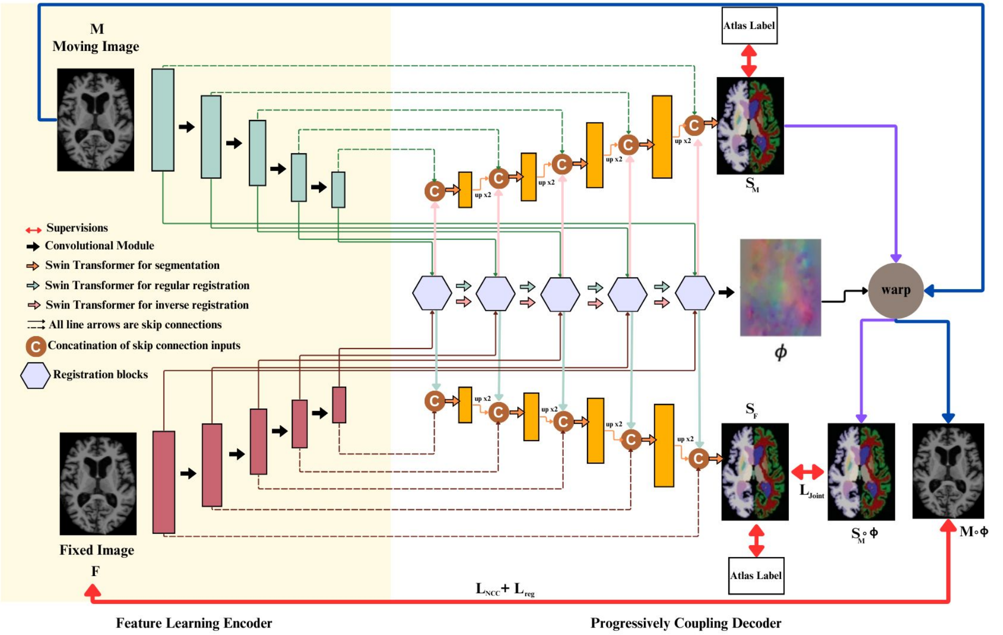
NICE-Transeg Model Architecture | NICE-Trans Encoder

Dual path intra-image feature learning encoder extracts image features from I_m and I_f

- Successive conv modules with 2x2 max pooling between adjacent modules
- Each conv module has two 3x3 conv layer followed by LeakyReLU with parameter 0.2

→ Learned features are repeatedly used in later registration and segmentation steps

→ Decouples feature learning from registration/segmentation decoding and avoids repeated feature learning when we perform multiple registration steps in the decoder

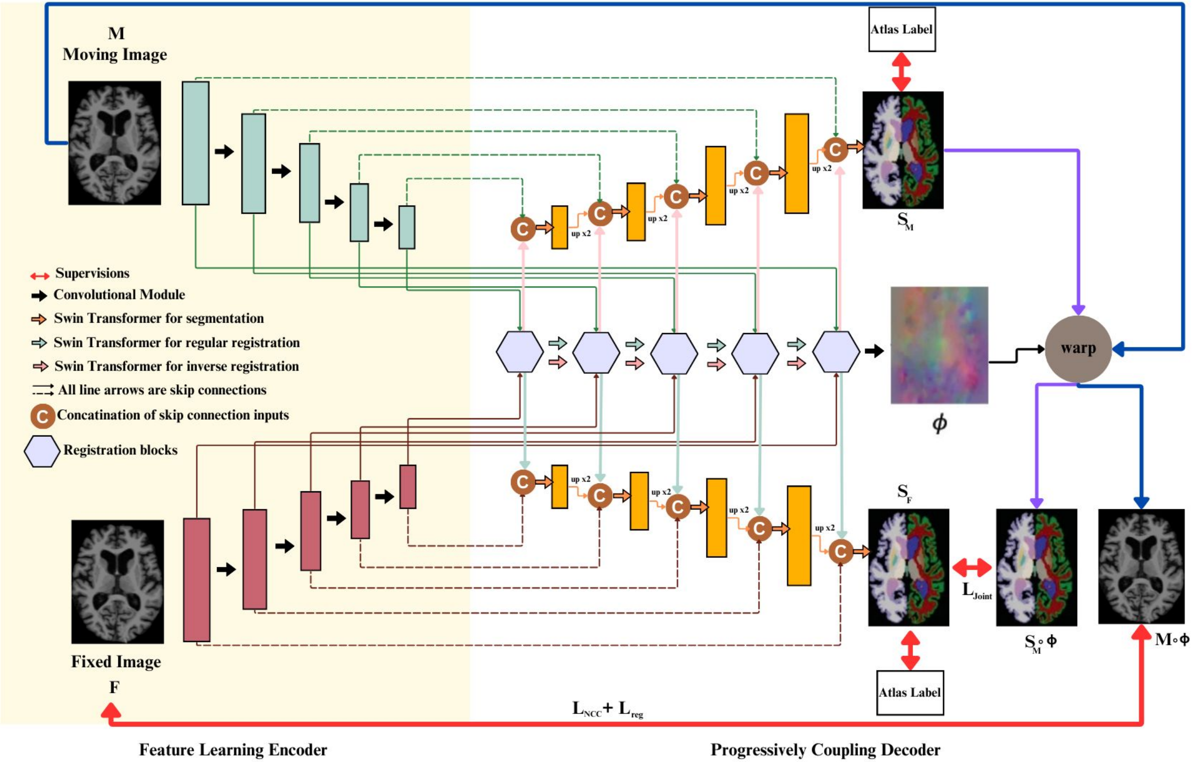

M
Moving Image
Atlas Label
S_M
up x2
C
Supervisions
Convolutional Module
Swin Transformer for segmentation
Swin Transformer for regular registration
Swin Transformer for inverse registration
All line arrows are skip connections
Concatination of skip connection inputs
Registration blocks
ϕ
warp
S_F
L_{Joint}
$S_M \circ \phi$
$M \circ \phi$
Fixed Image
F
Atlas Label
$L_{NCC} + L_{reg}$
Feature Learning Encoder
Progressively Coupling Decoder

NICE-Transeg Model Architecture | Decoders

NICE-Trans Registration Decoder Structure:

Joint Affine and Deformable Registration:
Generates 1 affine registration + 5 deformable registrations at dimensions equivalent to the respective encoder layers.

4 SwinTrans blocks + 3x3 and 1x1 convolutions


M
Moving Image
Atlas Label
S_M
Supervisions
Convolutional Module
Swin Transformer for segmentation
Swin Transformer for regular registration
Swin Transformer for inverse registration
All line arrows are skip connections
C Concatination of skip connection inputs
Registration blocks
up x2
C
ϕ
warp
S_F
L_{Joint}
$S_M \circ \phi$
$M \circ \phi$
Atlas Label
Fixed Image
F
$L_{NCC} + L_{reg}$
Feature Learning Encoder
Progressively Coupling Decoder

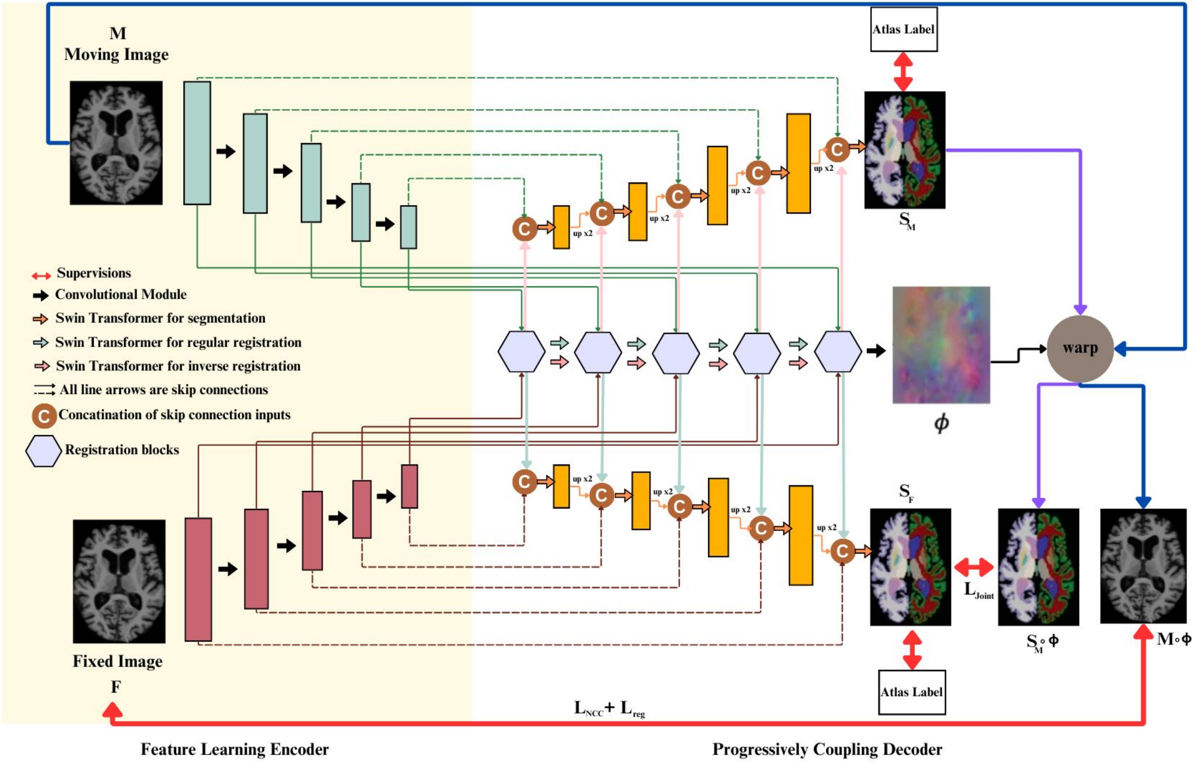
NICE-Transeg Model Architecture | Decoder

NICE-Trans Registration Decoder Structure:

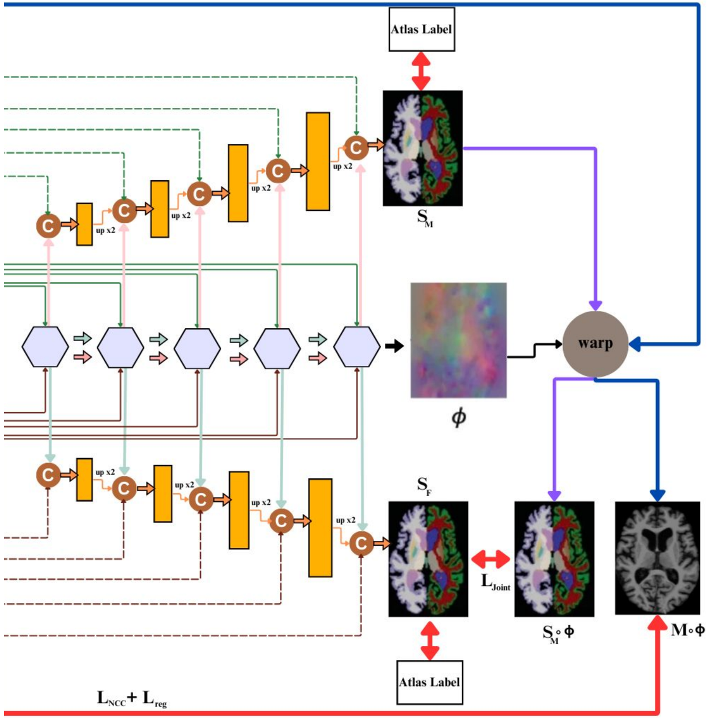
Joint Affine and Deformable Registration: Generates 1 affine registration + 5 deformable registrations at dimensions equivalent to the respective encoder layers.

New: Segmentation Branches for moving and fixed image

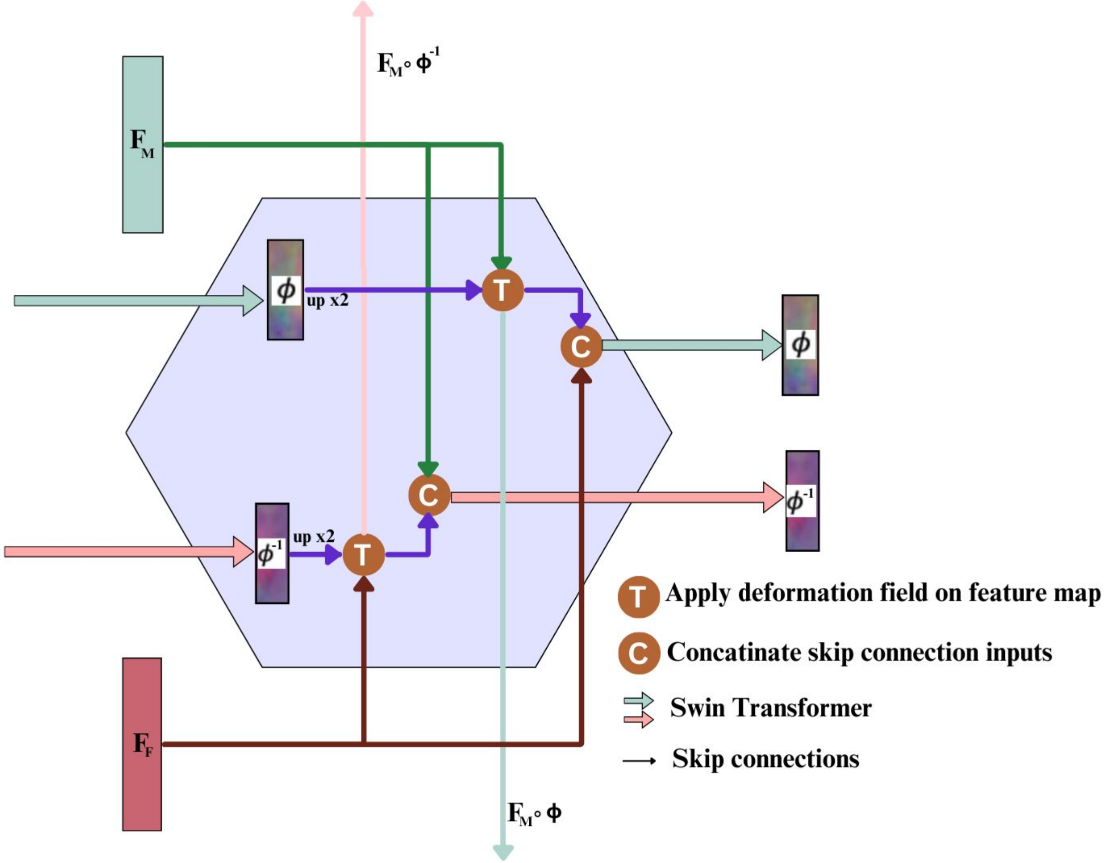
Another 4 SwinTrans blocks + 1x1 and 3x3 convolutions


M
Moving Image
Atlas Label
S_M
Supervisions
Convolutional Module
Swin Transformer for segmentation
Swin Transformer for regular registration
Swin Transformer for inverse registration
All line arrows are skip connections
C Concatination of skip connection inputs
Registration blocks
up x2
C
ϕ
warp
S_F
L_{Joint}
$S_M \circ \phi$
$M \circ \phi$
Fixed Image
F
$L_{NCC} + L_{reg}$
Atlas Label
Feature Learning Encoder
Progressively Coupling Decoder

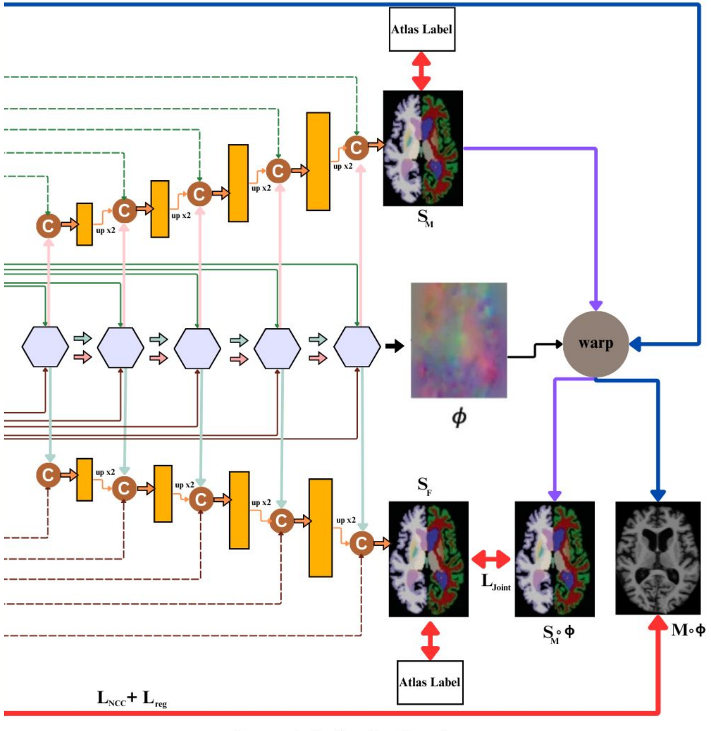
NICE-Transeg Model Architecture | "Progressively Coupling"



Progressively Coupling Decoder

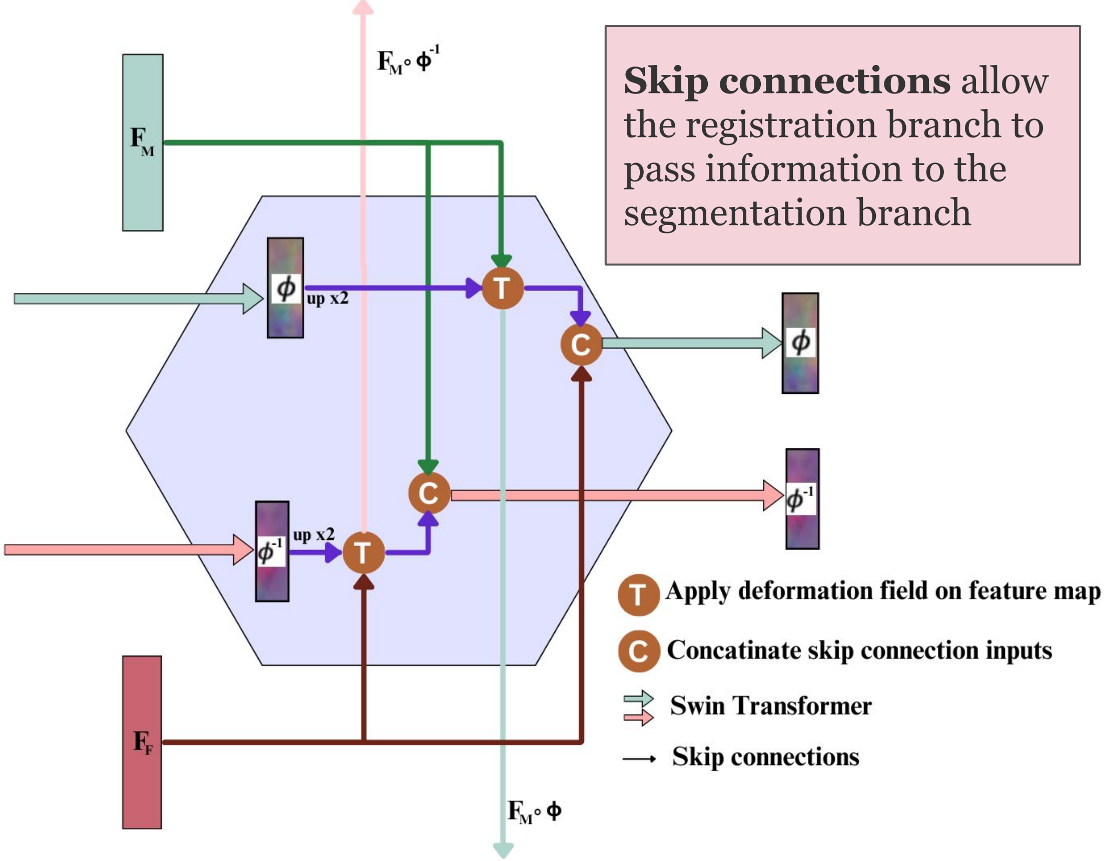


NICE-Transeg Model Architecture | “Progressively Coupling”



Progressively Coupling Decoder

Skip connections allow the registration branch to pass information to the segmentation branch



NICE-Transeg Model Architecture | Loss Calculation

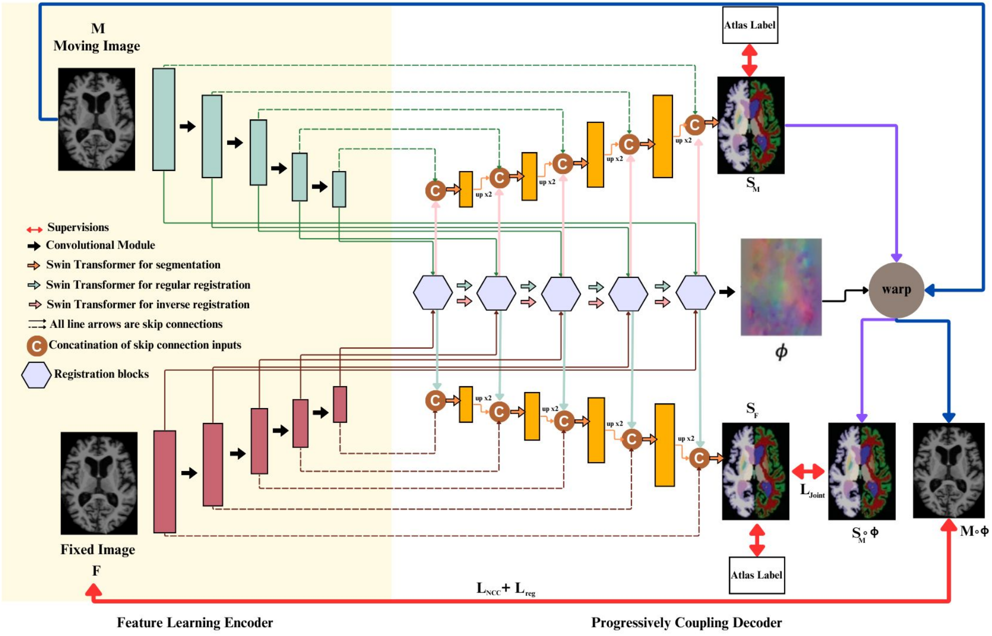
Unsupervised Learning in Registration Branch

$L = L_{sim} + \sigma L_{reg}$

L_{sim} = negative local normalized cross-correlation, which penalizes the differences between the warped image $I_{m \circ \phi}$ and the fixed image I_f

σ = regularization parameter (set to 1 in training)

L_{reg} = a diffusion regularizer (prevents overfitting)



M
Moving Image
Atlas Label
S_M
Supervisions
Convolutional Module
Swin Transformer for segmentation
Swin Transformer for regular registration
Swin Transformer for inverse registration
All line arrows are skip connections
C Concatination of skip connection inputs
Registration blocks
up x2
C
ϕ
warp
S_F
L_Joint
S_M∘ϕ
M∘ϕ
Fixed Image
F
Atlas Label
L_NCC + L_reg
Feature Learning Encoder
Progressively Coupling Decoder

NICE-Trans Model Architecture | Loss Calculation

Semi-supervised Learning in Segmentation Branch

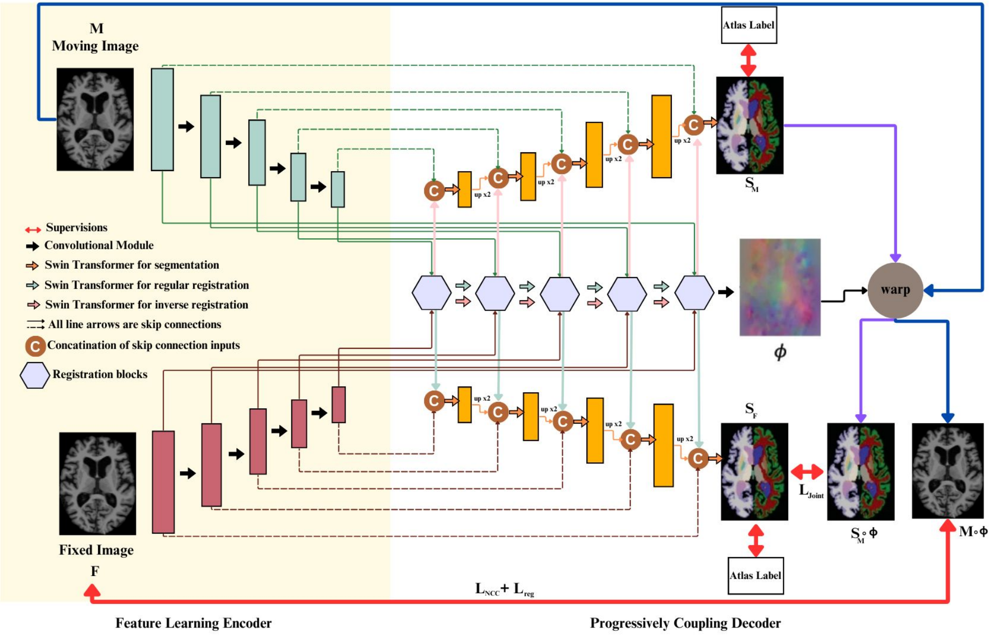
$L = L_{wce} + L_{dice} + L_{joint}$

L_{wce} = weighted cross entropy

L_{dice} = multiclass differentiable dice loss

$L_{joint} = L_{dice}(S_F , S_M \circ \phi)$

We do not use the segmentation labels of the training scans. Instead, only 5 labels of the atlas are used to simulate a **few-shot scenario**, mimicking the training procedure of PGCNet


M
Moving Image
Atlas Label
S_M
Supervisions
Convolutional Module
Swin Transformer for segmentation
Swin Transformer for regular registration
Swin Transformer for inverse registration
All line arrows are skip connections
C Concatination of skip connection inputs
Registration blocks
up x2
C
ϕ
warp
S_F
L_Joint
S_M∘ϕ
M∘ϕ
Fixed Image
F
Atlas Label
L_NCC + L_reg
Feature Learning Encoder
Progressively Coupling Decoder

Methodology | Dataset

Trained and tested the model on 2D **brain MRI registration and segmentation**

Experimented with **2** public datasets: 2D OASIS from HyperMorph and 2D IXI (generated from the 3d IXI dataset from TransMorph and made publicly available)

2D OASIS
160 x 192, skull stripped and bias-corrected norm images preprocessed with FreeSurfer,
25-label semantic segmentation maps

2D IXI
160 x 224, similarly processed with FreeSurfer, 256-label semantic segmentation maps (only 31 used)

Both datasets are split into training, validation, and testing (7:1:2)

5 training scans are randomly selected as Atlas from the training set.

Methodology | Implementation Detail

Pytorch on two RTX 3080s with 12 GB RAM each
Optimizer = ADAM with 1E-4 learning rate
Batch size = 1
All losses were weighted equally

150 epochs with early stopping

At each training iteration, one image is picked from the training set as the fixed image and one of the five atlas images is picked as the moving image, along with the ground truth atlas label map for the semi-supervised segmentation task.

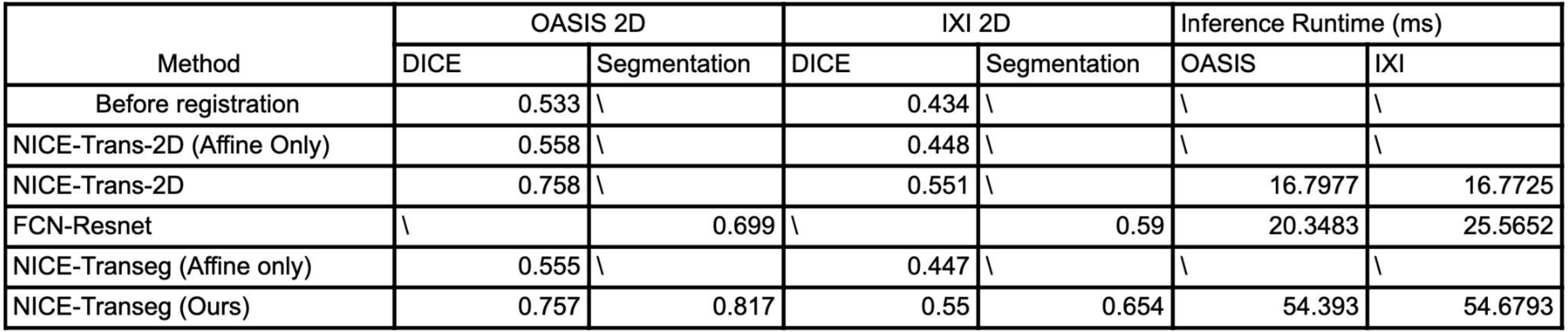
For validation and testing, two images are randomly picked from the corresponding split as the fixed and moving images

Methodology | Comparison Method

We compared the registration performance of NICE-Transeg with 2D NICE-Trans, which does not have the segmentation branches. **Dice similarity coefficient (DSC)** is used to measure registration accuracy.

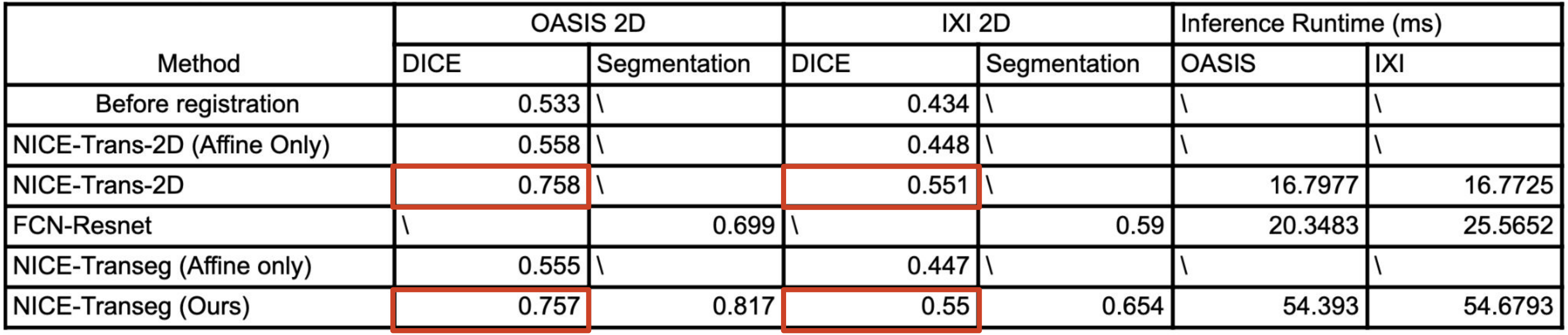
We also compare the few-shot semantic segmentation performance of NICE-Transeg with common semantic segmentation benchmark model FCN-ResNet. **Pixel-wise semantic segmentation accuracy** is measured to compare the result.

Results



Method	OASIS 2D		IXI 2D		Inference Runtime (ms)	
	DICE	Segmentation	DICE	Segmentation	OASIS	IXI
Before registration	0.533	\	0.434	\	\	\
NICE-Trans-2D (Affine Only)	0.558	\	0.448	\	\	\
NICE-Trans-2D	0.758	\	0.551	\	16.7977	16.7725
FCN-Resnet	\	0.699	\	0.59	20.3483	25.5652
NICE-Transeg (Affine only)	0.555	\	0.447	\	\	\
NICE-Transeg (Ours)	0.757	0.817	0.55	0.654	54.393	54.6793

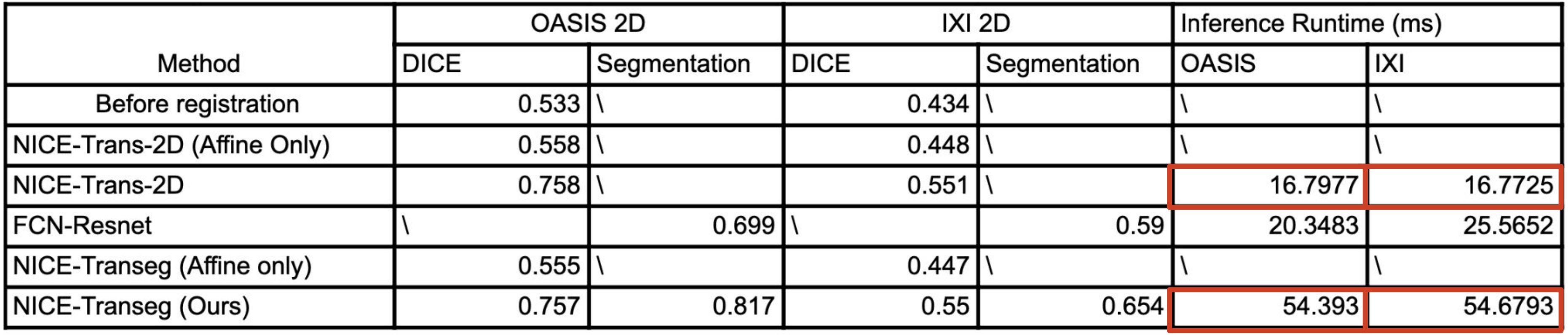
Results | Key Observations



Method	OASIS 2D		IXI 2D		Inference Runtime (ms)	
	DICE	Segmentation	DICE	Segmentation	OASIS	IXI
Before registration	0.533	\	0.434	\	\	\
NICE-Trans-2D (Affine Only)	0.558	\	0.448	\	\	\
NICE-Trans-2D	0.758	\	0.551	\	16.7977	16.7725
FCN-Resnet	\	0.699	\	0.59	20.3483	25.5652
NICE-Transeg (Affine only)	0.555	\	0.447	\	\	\
NICE-Transeg (Ours)	0.757	0.817	0.55	0.654	54.393	54.6793

1. NICE-Transeg had similar registration outcome for both OASIS and IXI when compared with NICE-Trans.

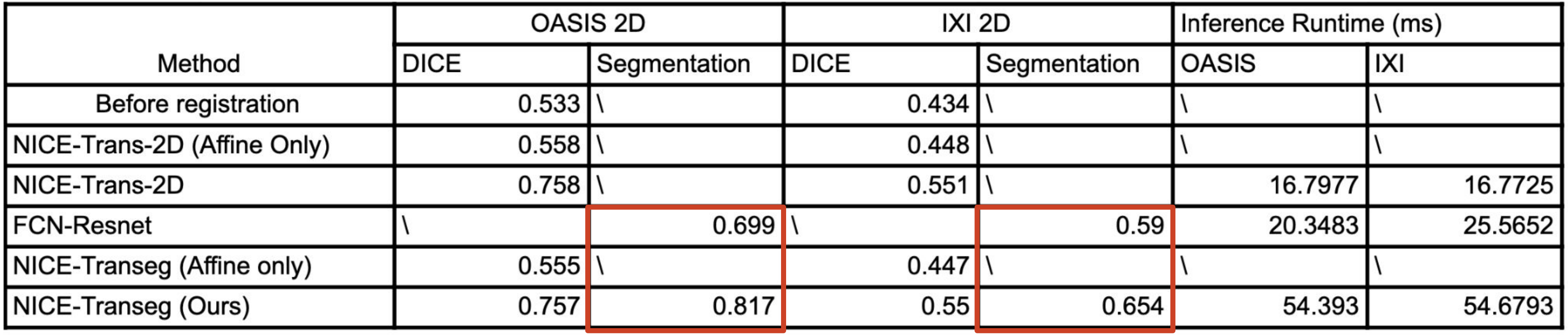
Results | Key Observations



Method	OASIS 2D		IXI 2D		Inference Runtime (ms)	
	DICE	Segmentation	DICE	Segmentation	OASIS	IXI
Before registration	0.533	\	0.434	\	\	\
NICE-Trans-2D (Affine Only)	0.558	\	0.448	\	\	\
NICE-Trans-2D	0.758	\	0.551	\	16.7977	16.7725
FCN-Resnet	\	0.699	\	0.59	20.3483	25.5652
NICE-Transeg (Affine only)	0.555	\	0.447	\	\	\
NICE-Transeg (Ours)	0.757	0.817	0.55	0.654	54.393	54.6793

2. Yet, the inference runtime for NICE-Transeg is significantly higher than NICE-Trans. This suggests that adding the segmentation branch is merely leading to higher computational cost without substantial benefits in the 2D registration setting.

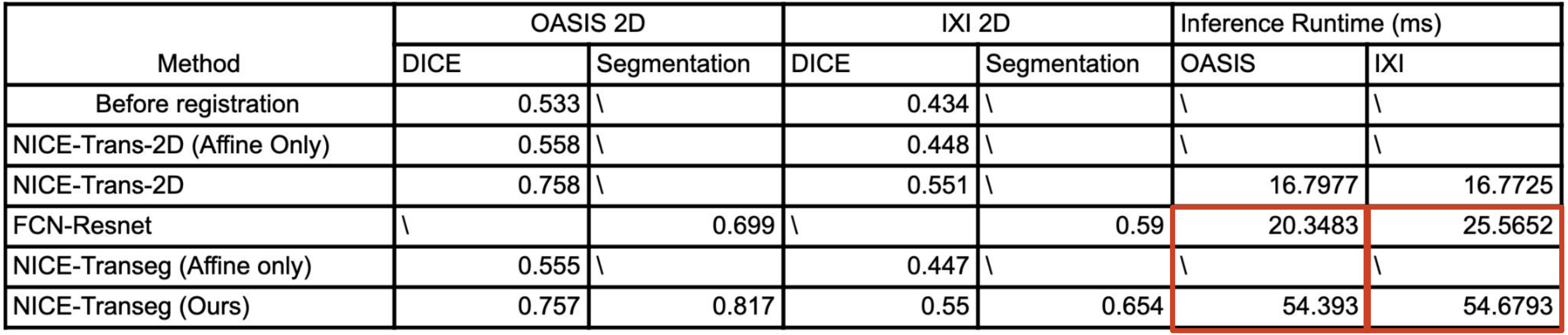
Results | Key Observations



Method	OASIS 2D		IXI 2D		Inference Runtime (ms)	
	DICE	Segmentation	DICE	Segmentation	OASIS	IXI
Before registration	0.533	\	0.434	\	\	\
NICE-Trans-2D (Affine Only)	0.558	\	0.448	\	\	\
NICE-Trans-2D	0.758	\	0.551	\	16.7977	16.7725
FCN-Resnet	\	0.699	\	0.59	20.3483	25.5652
NICE-Transeg (Affine only)	0.555	\	0.447	\	\	\
NICE-Transeg (Ours)	0.757	0.817	0.55	0.654	54.393	54.6793

3. NICE-Transeg obtained significant improvement(0.118 and 0.064) in segmentation outcome when compared to FCN-Resnet. Both of these models were trained on only 5 ground truth label maps

Results | Key Findings



Method	OASIS 2D		IXI 2D		Inference Runtime (ms)	
	DICE	Segmentation	DICE	Segmentation	OASIS	IXI
Before registration	0.533	\	0.434	\	\	\
NICE-Trans-2D (Affine Only)	0.558	\	0.448	\	\	\
NICE-Trans-2D	0.758	\	0.551	\	16.7977	16.7725
FCN-Resnet	\	0.699	\	0.59	20.3483	25.5652
NICE-Transeg (Affine only)	0.555	\	0.447	\	\	\
NICE-Transeg (Ours)	0.757	0.817	0.55	0.654	54.393	54.6793

4. However, NICE-Transeg also requires a higher inference runtime in comparison to FCN-Resnet

Discussion | Conclusion & Future Directions

Summary of our key findings and contributions:

1. NICE-Trans 2D did not benefit from the progressive coupling of segmentation and registration branches.
2. NICE-Transeg showed significant improvement in 2D semantic segmentation performance over baseline convolutional model FCN-Resnet.
3. We created the 2D version of IXI and made it publicly available as an open source dataset.

Discussion | Conclusion & Future Directions

Summary of our key findings and contributions:

1. NICE-Trans 2D did not benefit from the progressive coupling of segmentation and registration branches.
2. NICE-Transeg showed significant improvement in 2D semantic segmentation performance over baseline convolutional model FCN-Resnet.
3. We created the 2D version of IXI and made it publicly available as an open source dataset.

Future Directions

1. Investigate the performance of 3D NICE-Transeg – it is possible that 3D image registration may benefit more from the segmentation branches because it is a much more challenging task compared to 2D image registration.
2. Does NICE-Transeg produce **more realistic** results?
3. Compare with more few-shot semantic segmentation benchmarks to further assess how NICE-Transeg performs compared to SOTA
4. Incorporate positional embeddings proposed in PGCNet into NICE-Transeg

Questions?