Sahin Olut

- Second-year PhD student in UNC-CH.
- Worked one year in industry before PhD.
- Published papers in ECCV, MICCAI, MIDL.
- Have experience on **image registration**, **image segmentation**, **image-to-image synthesis** with deep learning.

Visualization of Tourist Trends

- Problem: Many unlabeled images from images at Instagram, representing various tourist activities.
- Goal: We would like to cluster those images where similar activities are clustered together.
- Solution: Use deep networks to encode this information to the vectors and cluster those vectors.

Overall Pipeline

Deep Network

[Feature vectors]

tSNE

Live Demo:

- Click!

Why tSNE and not PCA?

- PCA is limited to linear relations. (We certainly have non-linear relations)

- PCA works globally, where tSNE accounts for both local and global relations. (It makes it easier to handle many images)

- tSNE maps the high dimensional distribution to a lower dimensional distribution, tries to minimize the distance between them.

Adversarial Data Augmentation via Deformation Statistics

Sahin Olut, Zhengyang Shen, Zhenlin Xu,
Samuel Gerber, Marc Niethammer

UNC Chapel Hill and Kitware

ECCV'20 ONLINE
23-28 AUGUST 2020

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

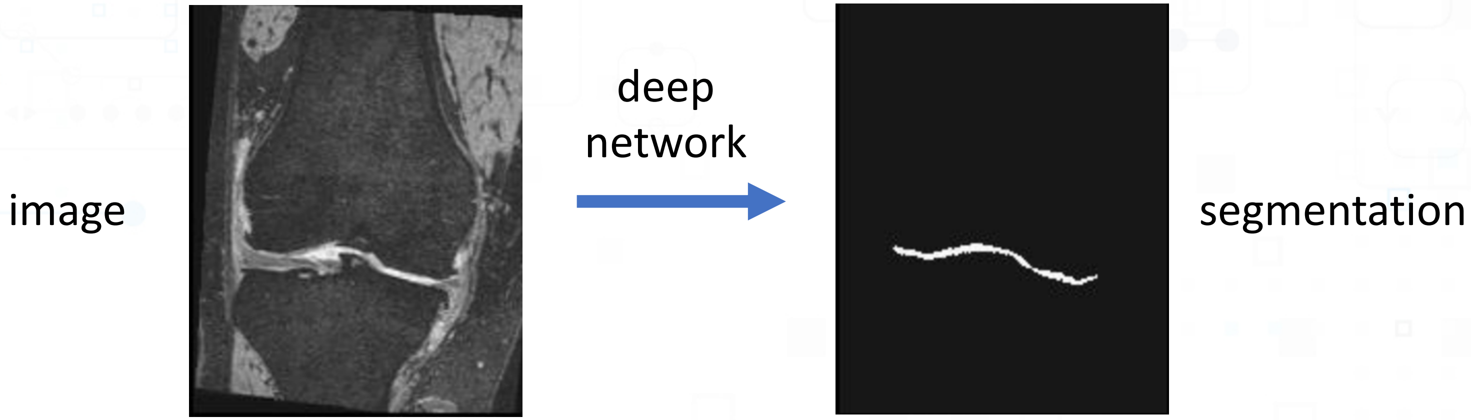
Kitware

Problem

Goal: Medical image segmentation with deep learning.

Problem: Acquiring labeled medical data is labor-intensive.

We need additional data to train a deep segmentation network.

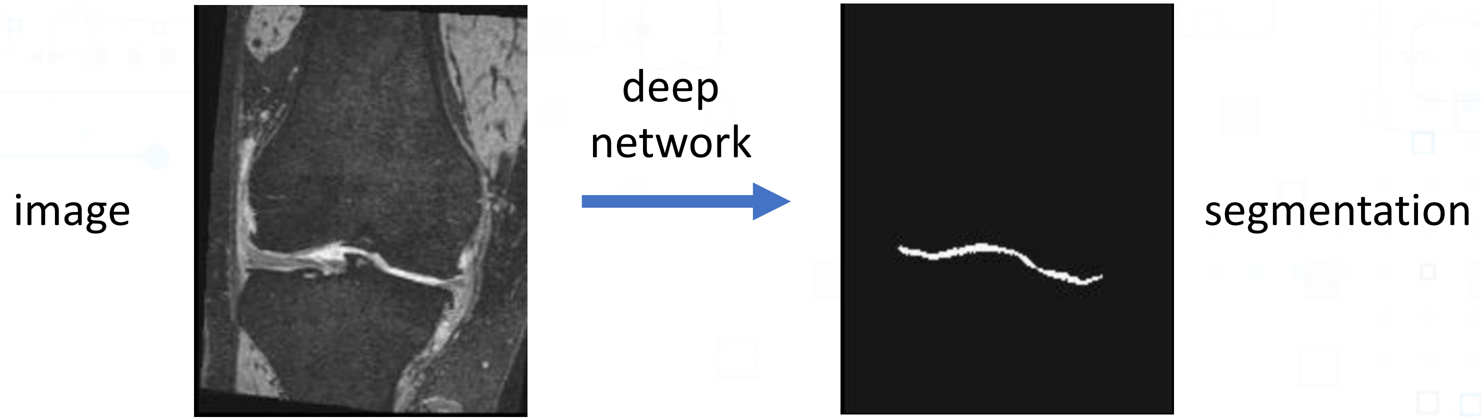


Problem

Goal: Medical image segmentation with deep learning.

Problem: Acquiring labeled medical data is labor-intensive.

1st Solution: Use unlabeled images to model *realistic augmentations*.

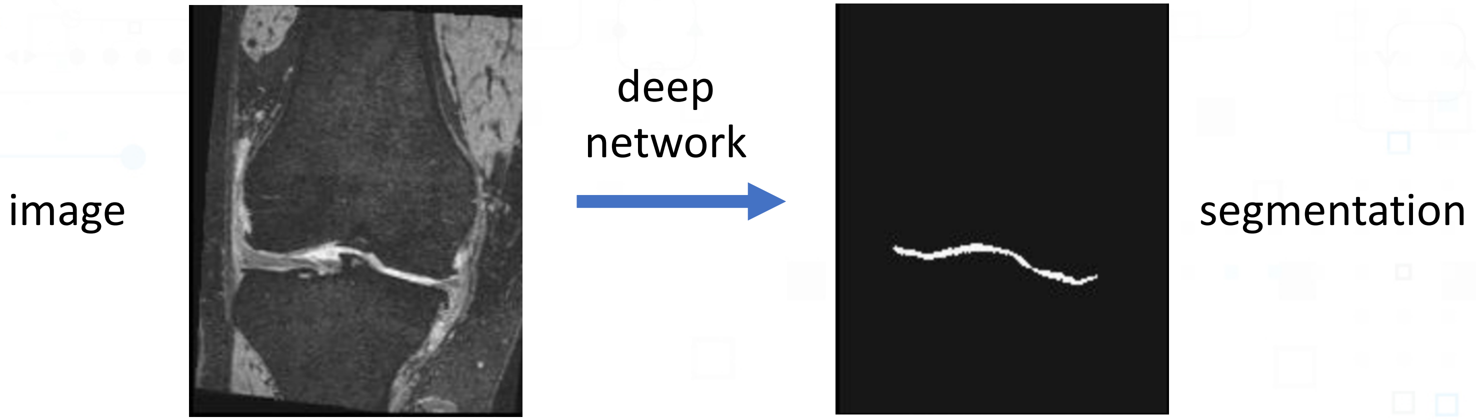


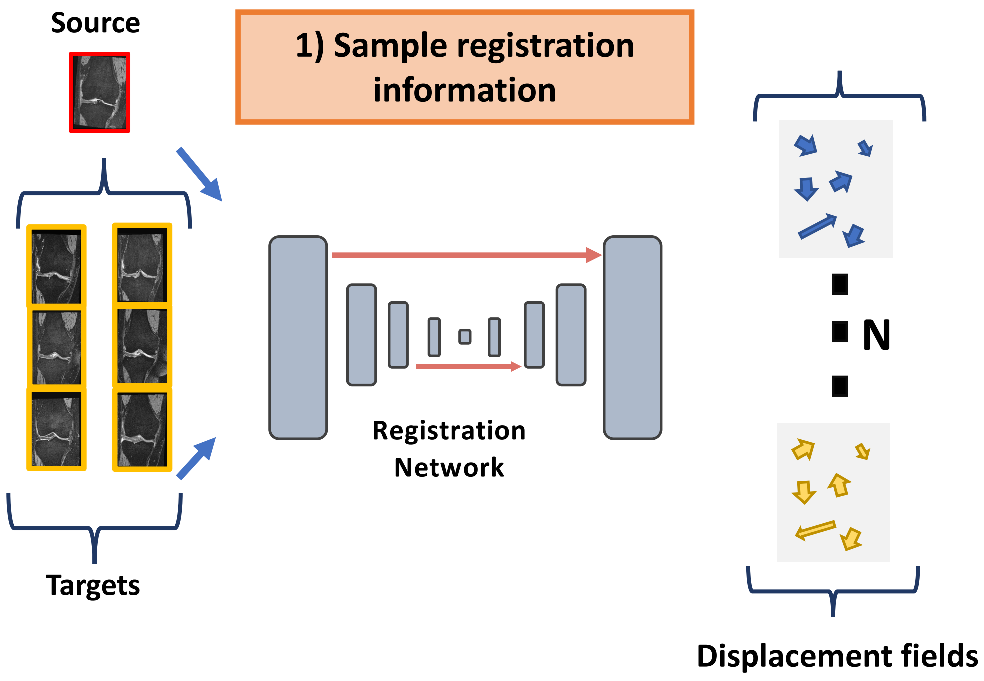
Problem

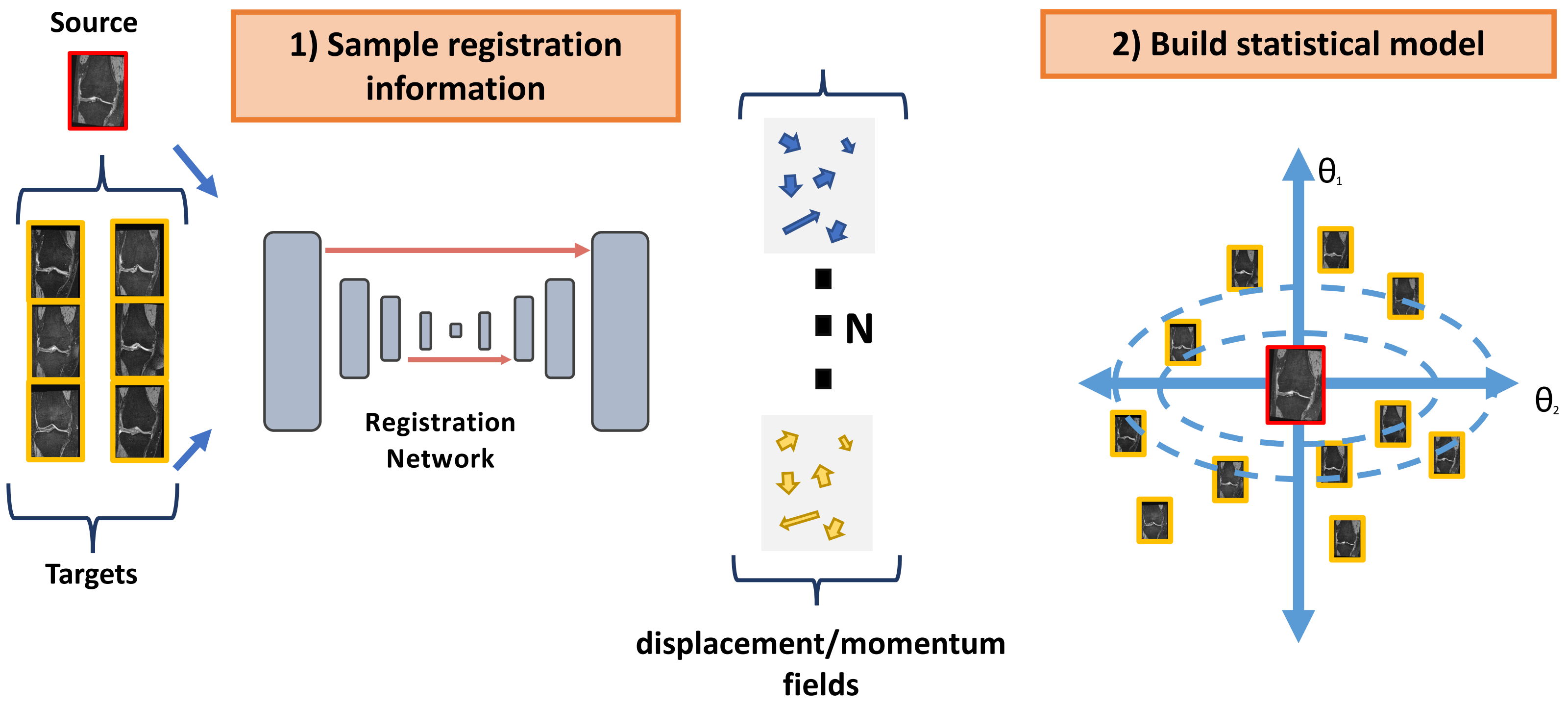
Goal: Medical image segmentation with deep learning.

Problem: Acquiring labeled medical data is labor-intensive.

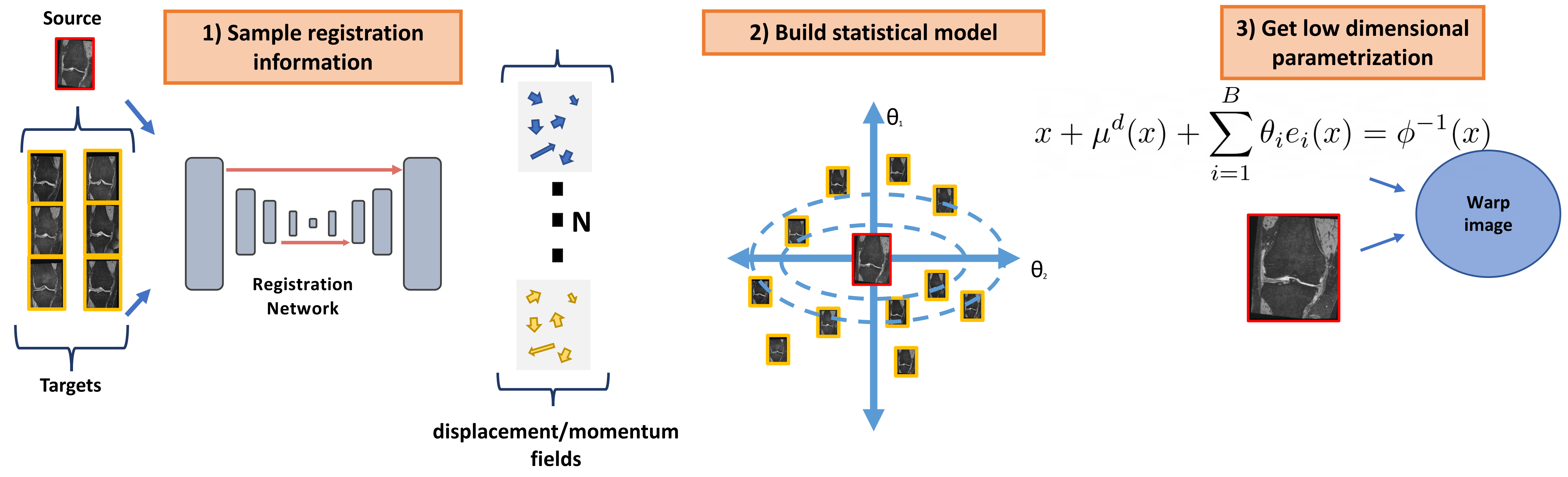
2nd Solution: Use adversarial data augmentation to obtain additional challenging training samples.

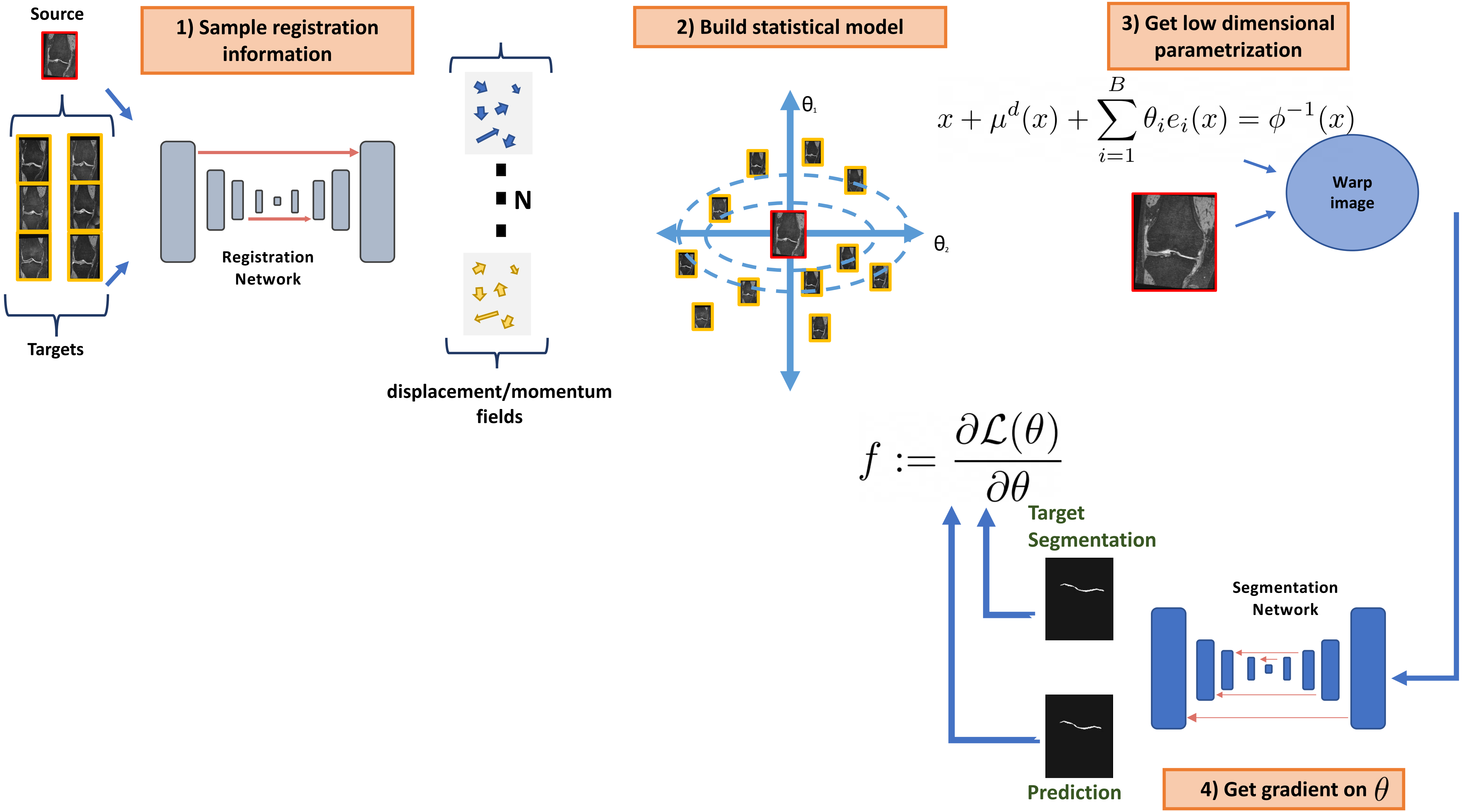



Source
1) Sample registration information
Registration Network
N
Targets
Displacement fields

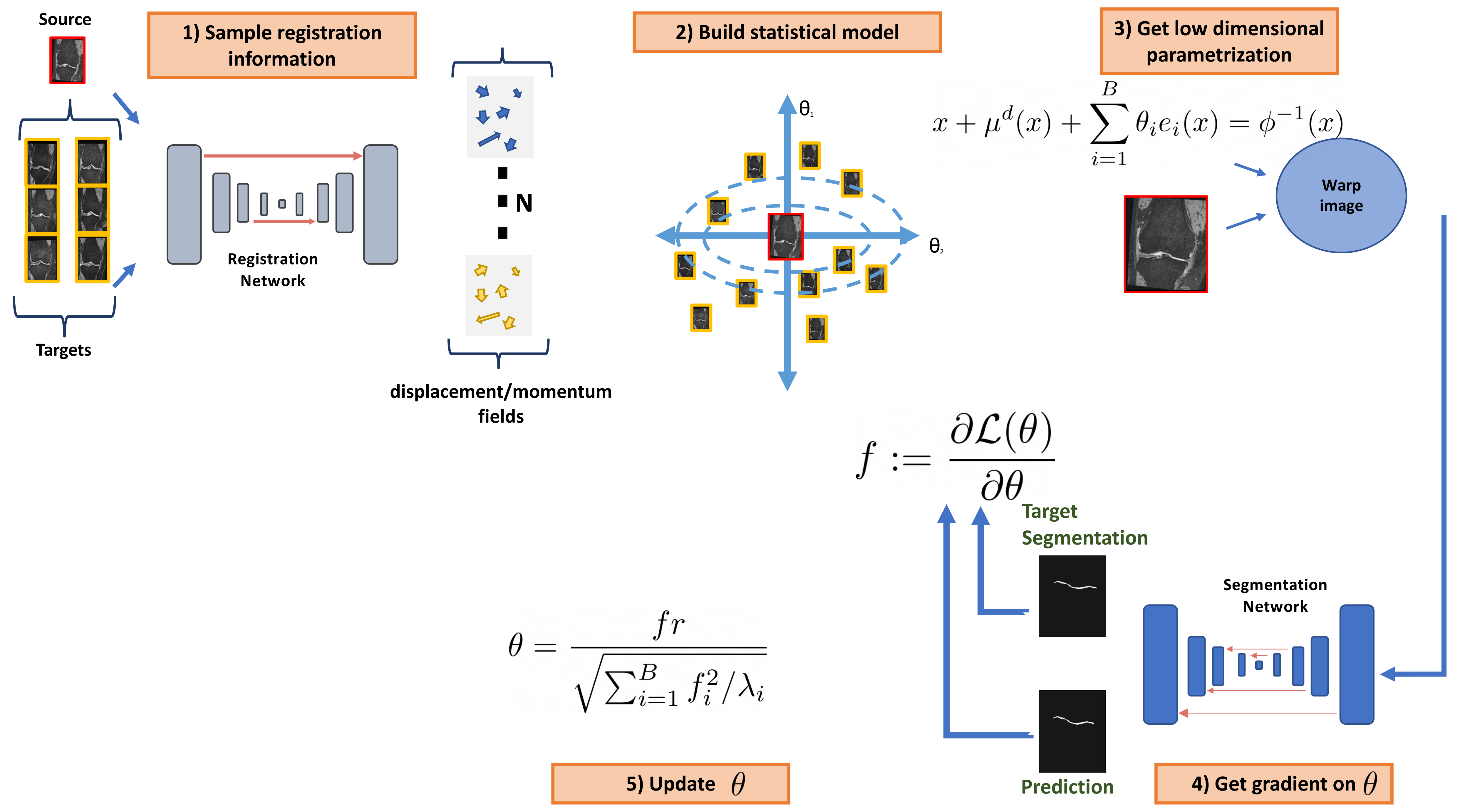


Source
1) Sample registration information
2) Build statistical model
Targets
Registration Network
N
displacement/momentum fields
θ1
θ2

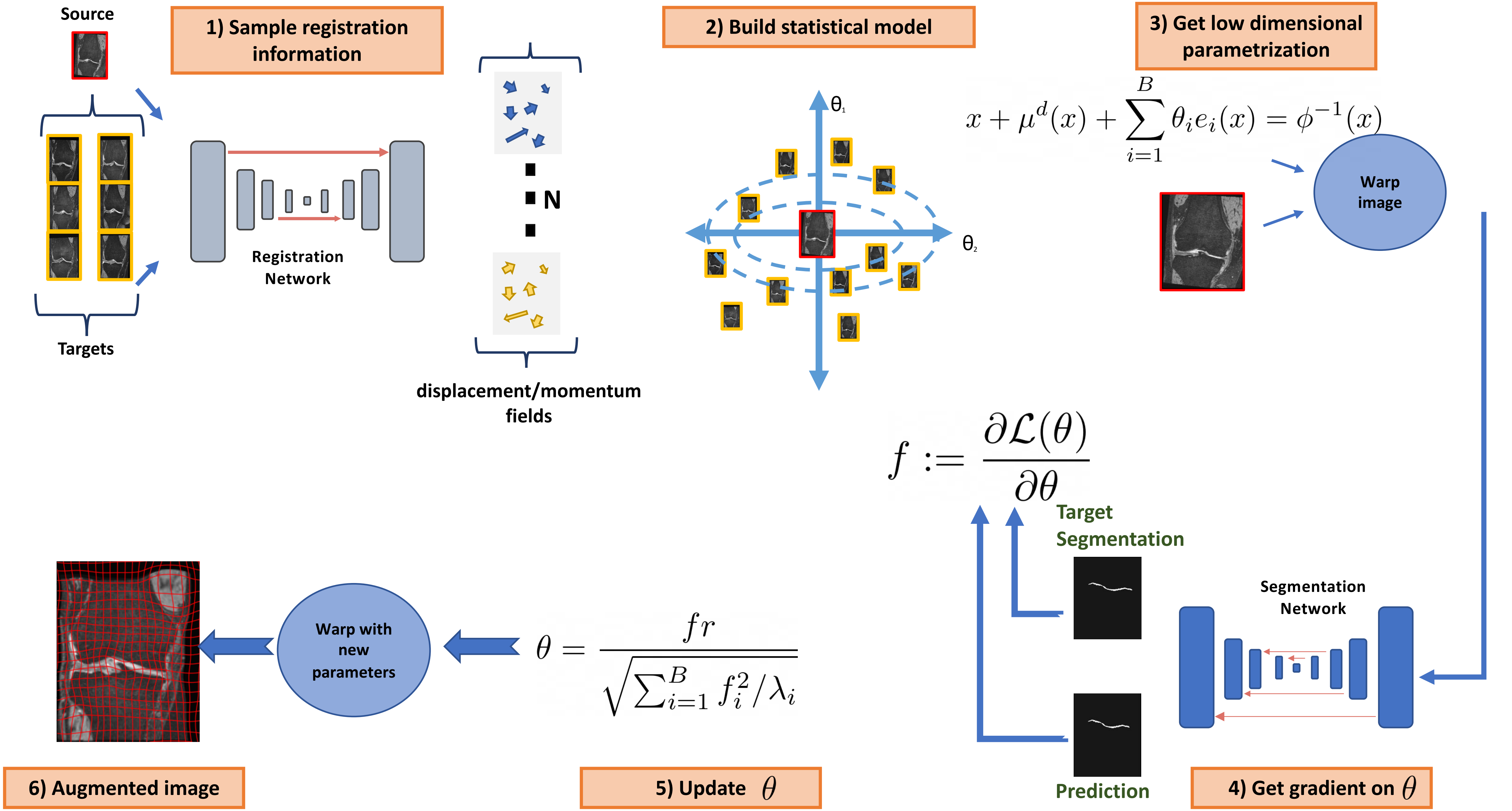

Source
1) Sample registration information
Registration Network
Targets
N
displacement/momentum fields
2) Build statistical model
θ1
θ2
3) Get low dimensional parametrization
$x + \mu^d(x) + \sum_{i=1}^{B} \theta_i e_i(x) = \phi^{-1}(x)$
Warp image



Source
1) Sample registration information
Registration Network
Targets
N
displacement/momentum fields
2) Build statistical model
θ1
θ2
3) Get low dimensional parametrization
$x + \mu^d(x) + \sum_{i=1}^{B} \theta_i e_i(x) = \phi^{-1}(x)$
Warp image
$f := \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$
Target Segmentation
Segmentation Network
Prediction
4) Get gradient on θ



Source
1) Sample registration information
Targets
Registration Network
N
displacement/momentum fields
2) Build statistical model
θ1
θ2
3) Get low dimensional parametrization
$x + \mu^d(x) + \sum_{i=1}^{B} \theta_i e_i(x) = \phi^{-1}(x)$
Warp image
$f := \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$
Target Segmentation
Segmentation Network
Prediction
4) Get gradient on θ
$\theta = \frac{fr}{\sqrt{\sum_{i=1}^{B} f_i^2/\lambda_i}}$
5) Update θ



Source
1) Sample registration information
2) Build statistical model
3) Get low dimensional parametrization
$x + \mu^d(x) + \sum_{i=1}^{B} \theta_i e_i(x) = \phi^{-1}(x)$
Warp image
Registration Network
N
θ_1
θ_2
Targets
displacement/momentum fields
$f := \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$
Target Segmentation
Segmentation Network
Warp with new parameters
$\theta = \frac{fr}{\sqrt{\sum_{i=1}^{B} f_i^2/\lambda_i}}$
6) Augmented image
5) Update θ
Prediction
4) Get gradient on θ

Our contributions

Statistical Deformation Model: We estimate image-specific statistical deformation models via PCA for realistic augmentations.

Efficient Estimation: We efficiently estimate these image-specific statistical deformation models via deep registration networks.

Adversarial strategy: We use our statistical deformation models within an adversarial strategy to obtain challenging, realistic deformations.

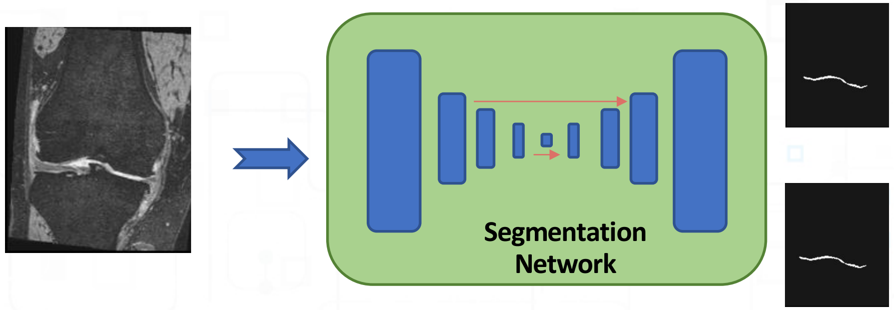
Adversarial affine transformations

We can create new adversarial augmentations by affine transformations [3].

$$Ax + b = y$$

Apply affine transformation

Prediction ($\hat{y}$)



Manual
Segmentation (y)

$$\mathcal{L}(A, b) = loss(y, \hat{y})$$

$$\frac{\partial \mathcal{L}(A,b)}{\partial A}, \frac{\partial \mathcal{L}(A,b)}{\partial b}$$

Gradient ascent on A, b

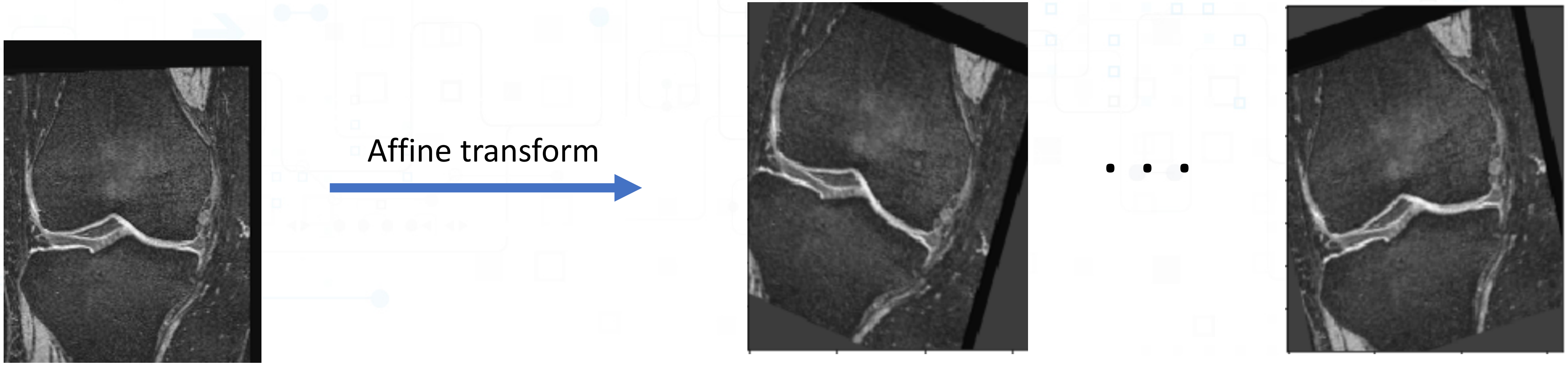
Update affine transformation
parameters in gradient direction

[3] Engstrom etl al., “A rotation and a translation suffice: Fooling CNNs with simple transformations”. arXiv preprint arXiv:1712.02779

Shortcomings of affine transformations

We can create new adversarial augmentations by affine transformations.

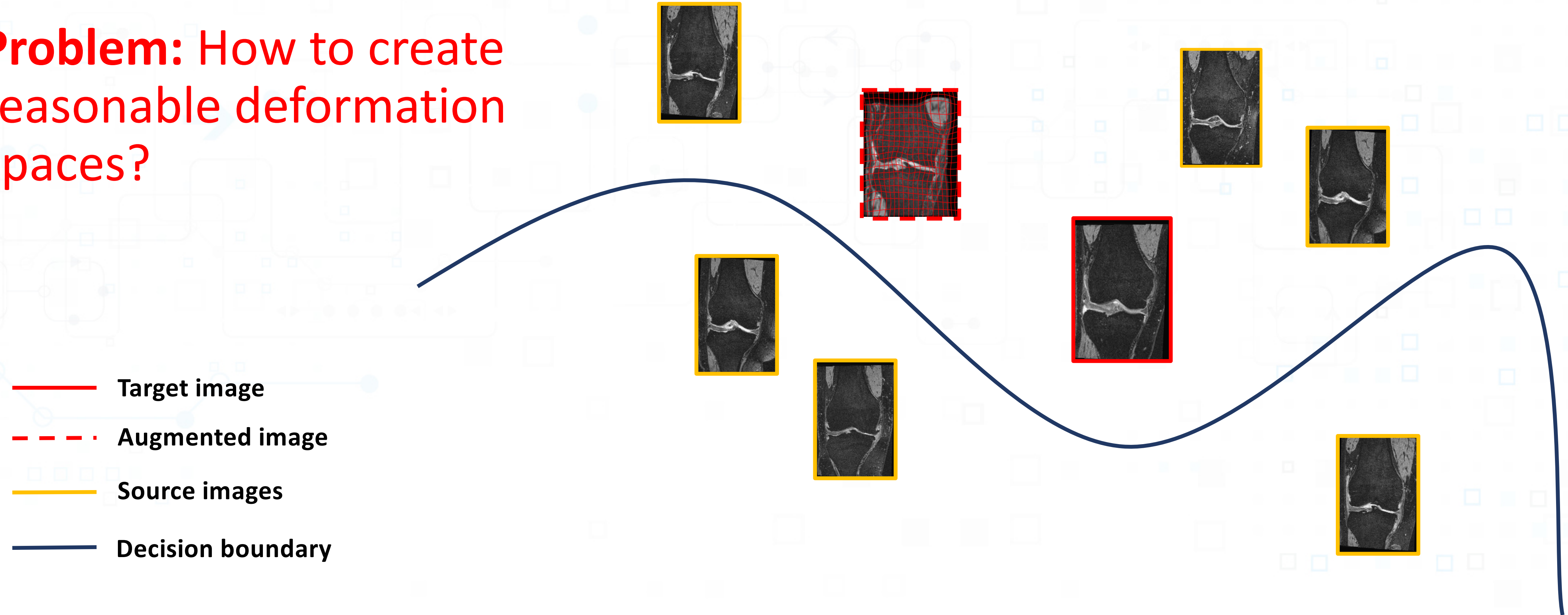
Problem: Affine transformations are too simple to model realistic deformations.



Our adversarial training

Creating realistic adversarial images: We augment our images in deformation space for better realism.

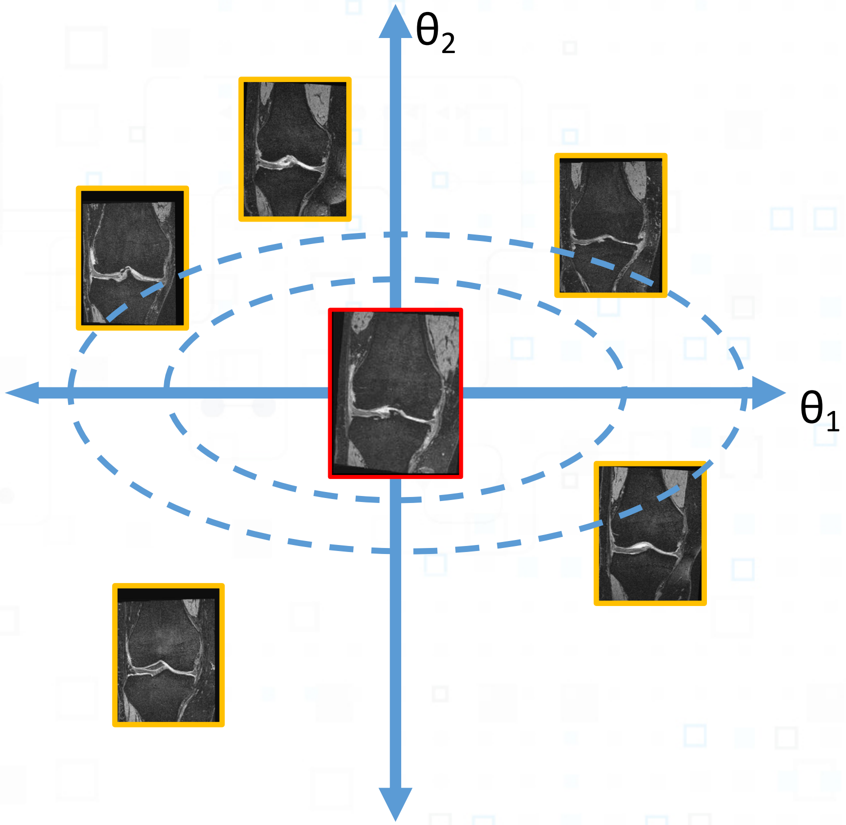
Problem: How to create reasonable deformation spaces?



Statistical deformation model

Problem: How to create reasonable deformation spaces?

Solution: We build an image-specific statistical deformation model from sets of deformations.

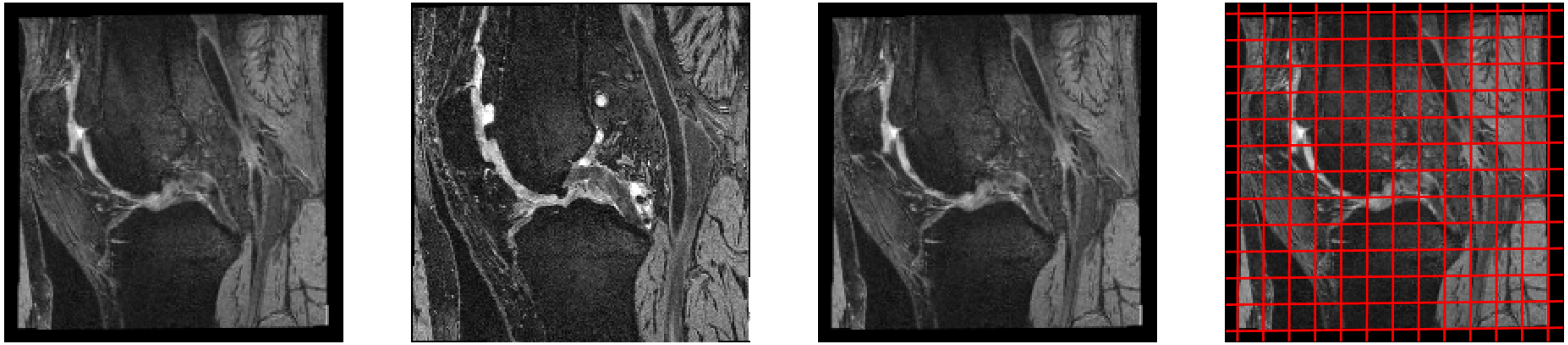


This model can be built with **displacement fields** or **initial momentum fields**.

Sets of deformations help us to define dataset variations.

Image registration

We use **pair-wise registrations to define a statistical model** to understand variations of deformations.



source target deformed

Estimating deformations quickly

Problem: Estimating a set of deformations with image registration is costly.

Solution: We quickly obtain deformations with a deep registration network.

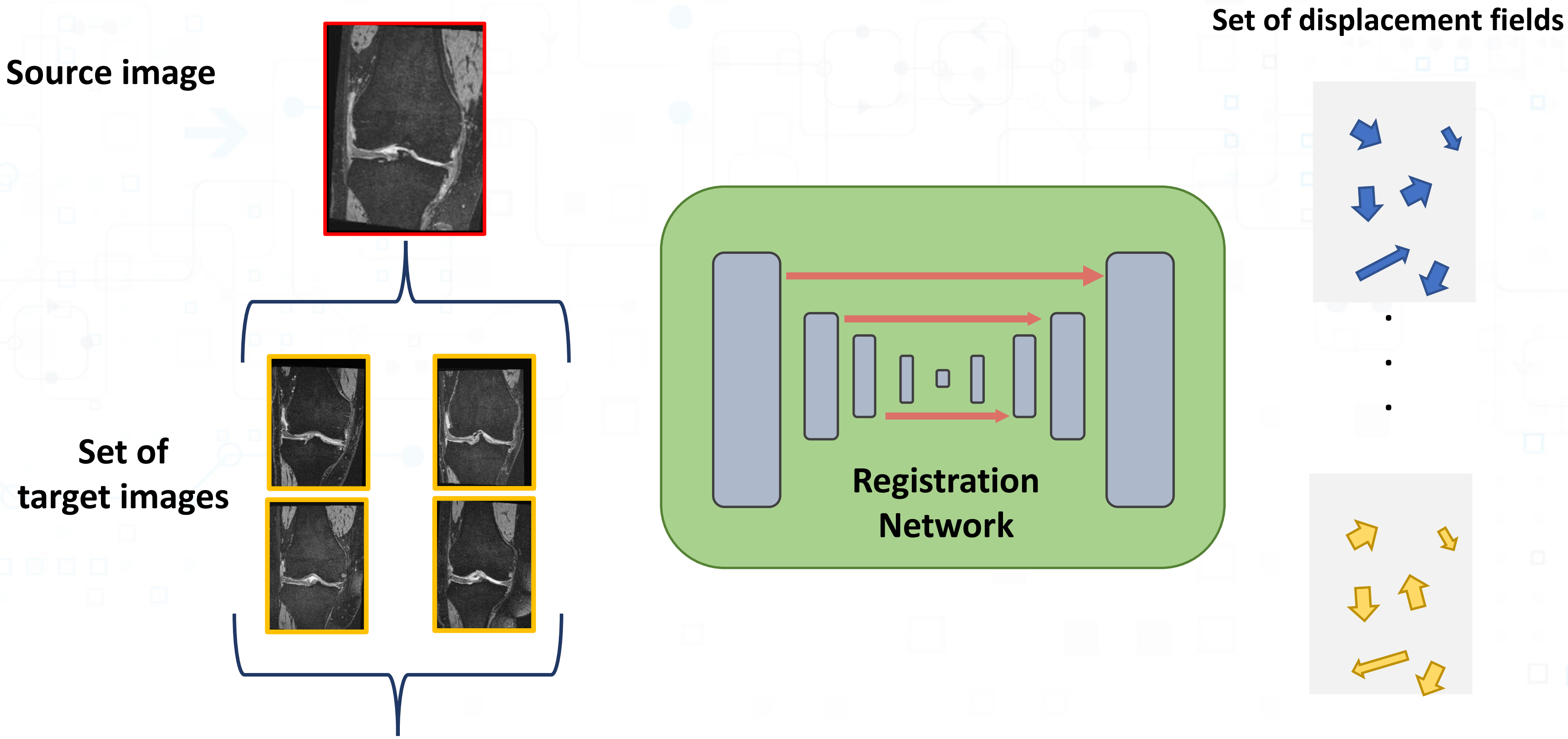
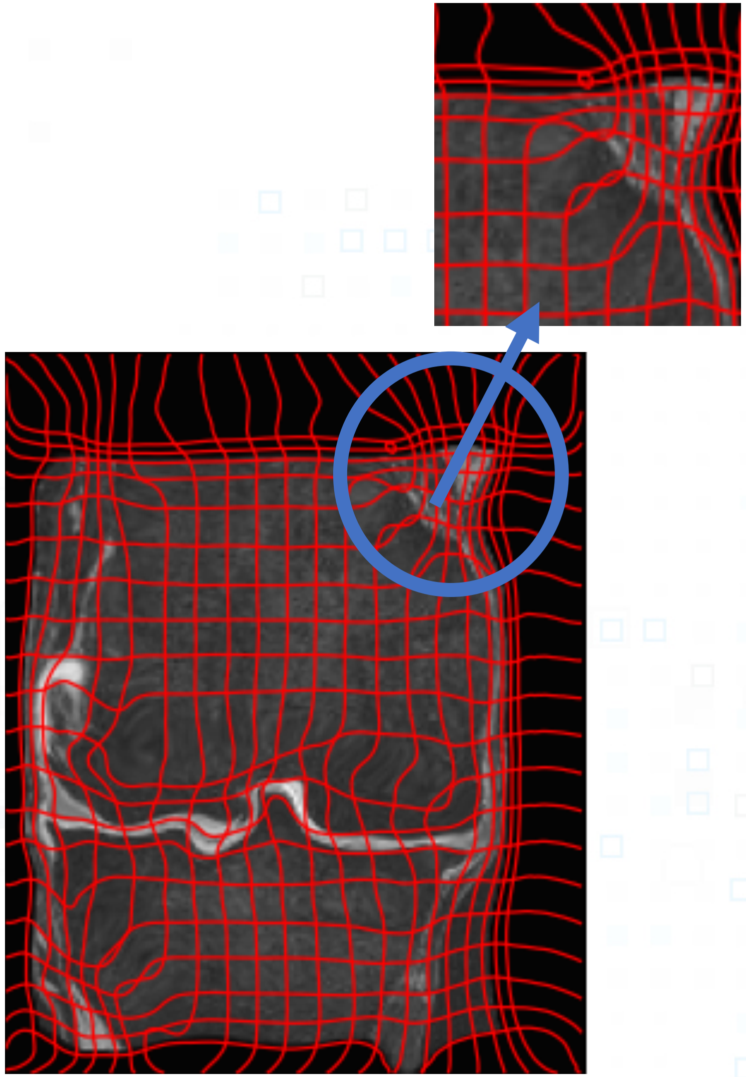


Image registration

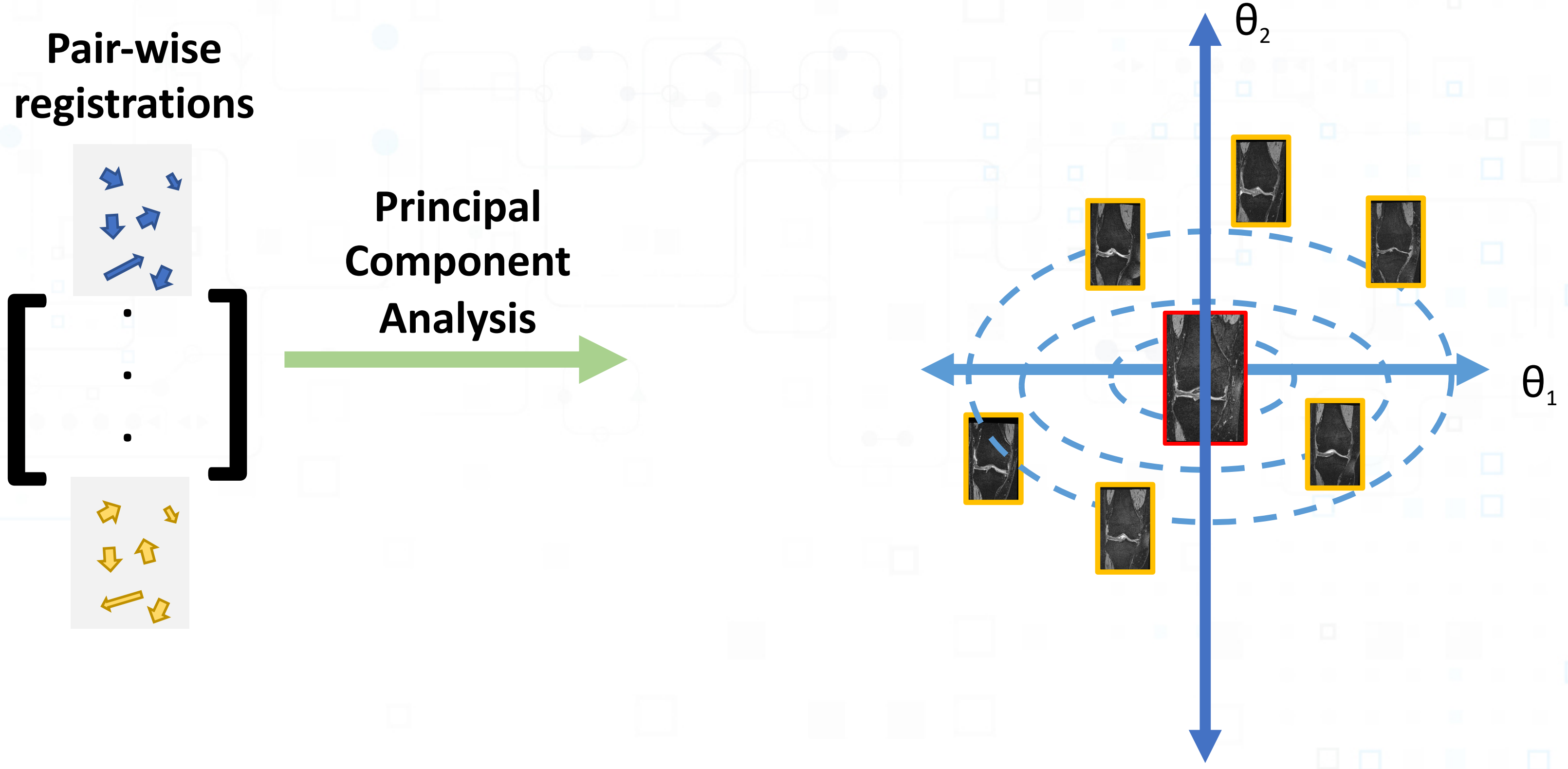
Displacement field prediction: We directly predict a displacement field via a deep neural network.

Foldings may occur in case of large variations in the deformation model.



Statistical deformation model

1) We obtain pair-wise registrations for a source image which is to be augmented.



Statistical deformation model

2) More specifically, we use PCA to obtain a low dimensional deformation parametrization.

Coefficients parametrizing deformation space

$$\phi^{-1}(x) \leftarrow x + \mu^d(x) + \sum_{i=1}^{B} \theta_i e_i(x)$$

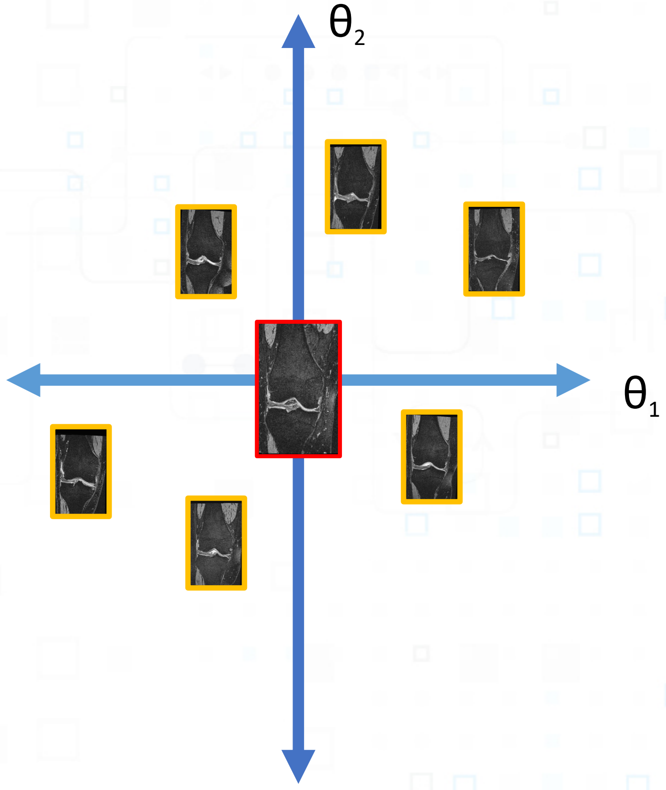
Deformation map

Mean displacement

Basis for deformation space

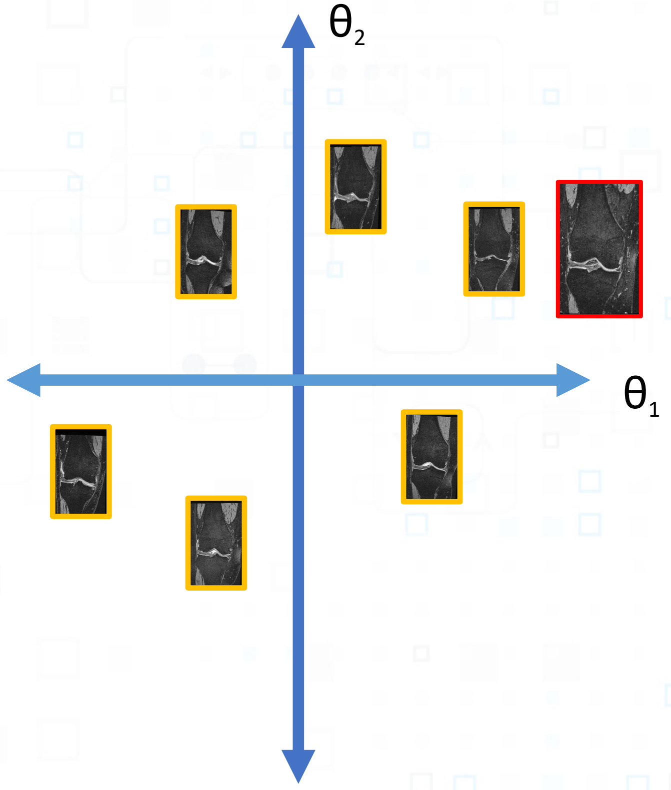
Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.



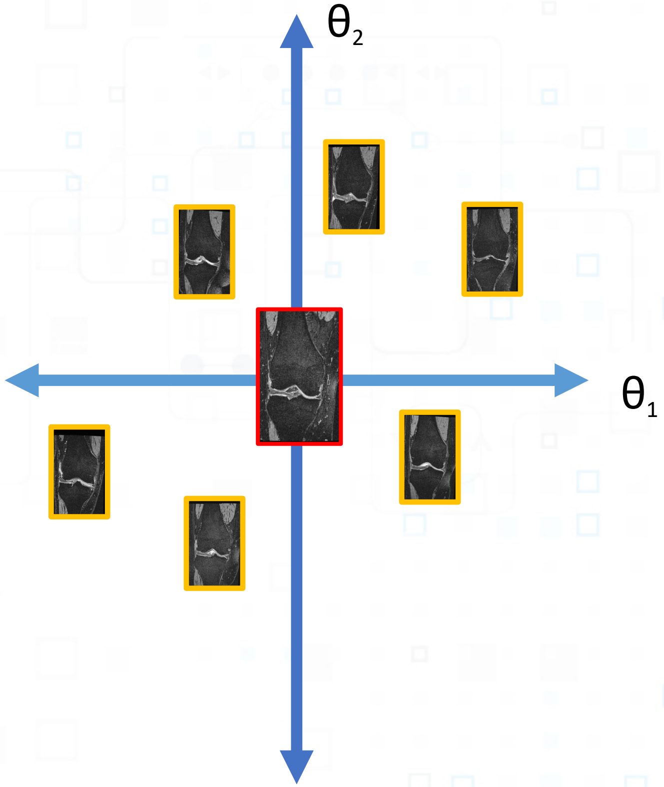
Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.



Statistical deformation model

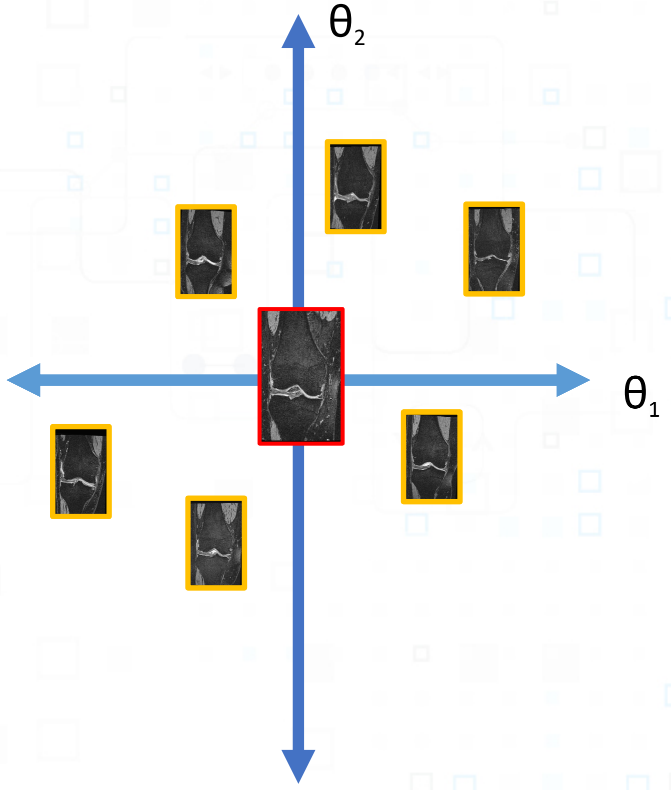
Our PCA model defines an **image-specific** statistical deformation space.



Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.

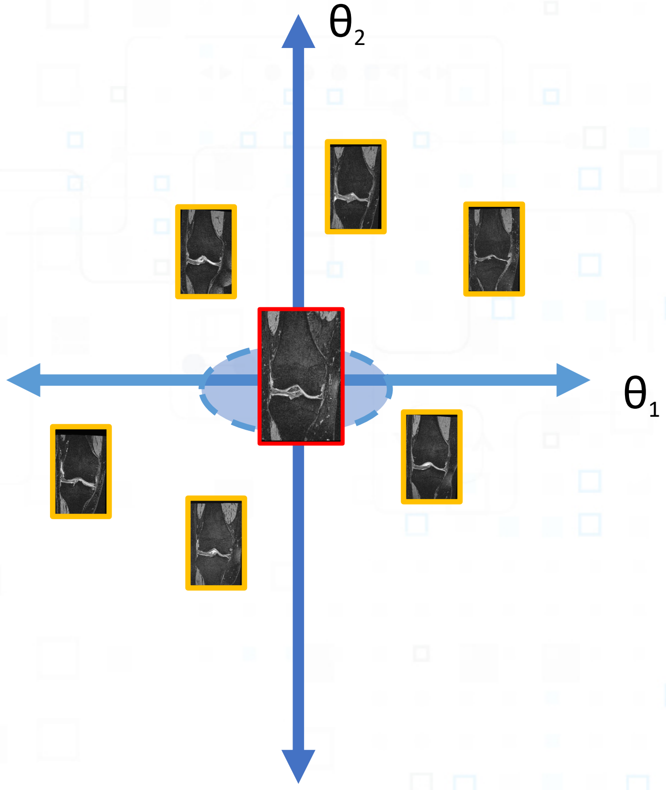
1st problem: How far should we move the source image?



Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.

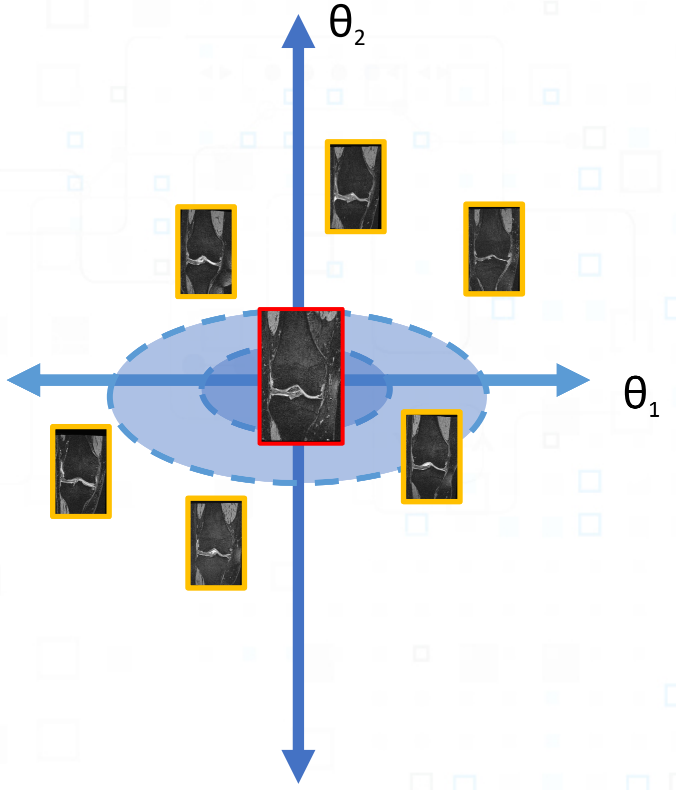
1st problem: How far should we move the source image?



Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.

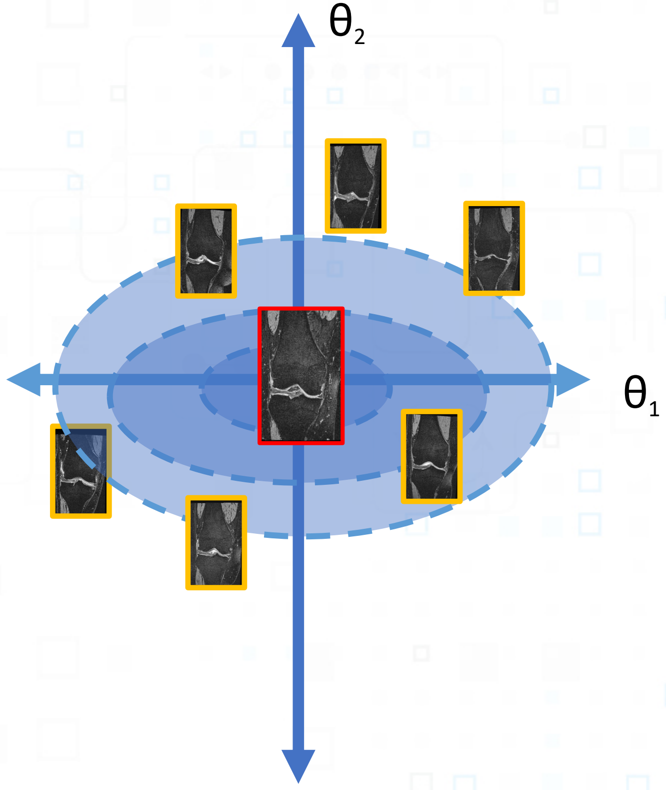
1st problem: How far should we move the source image?



Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.

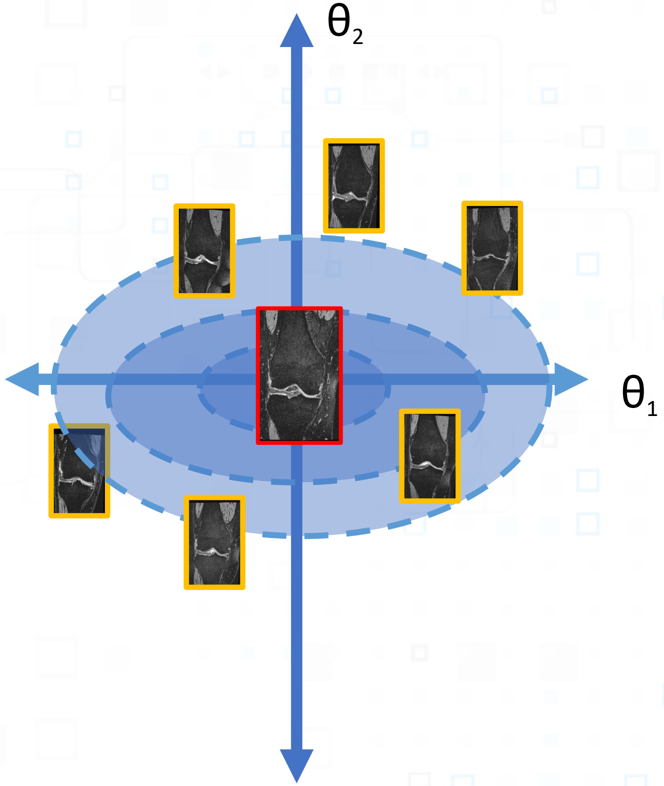
1st problem: How far should we move the source image?



Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.

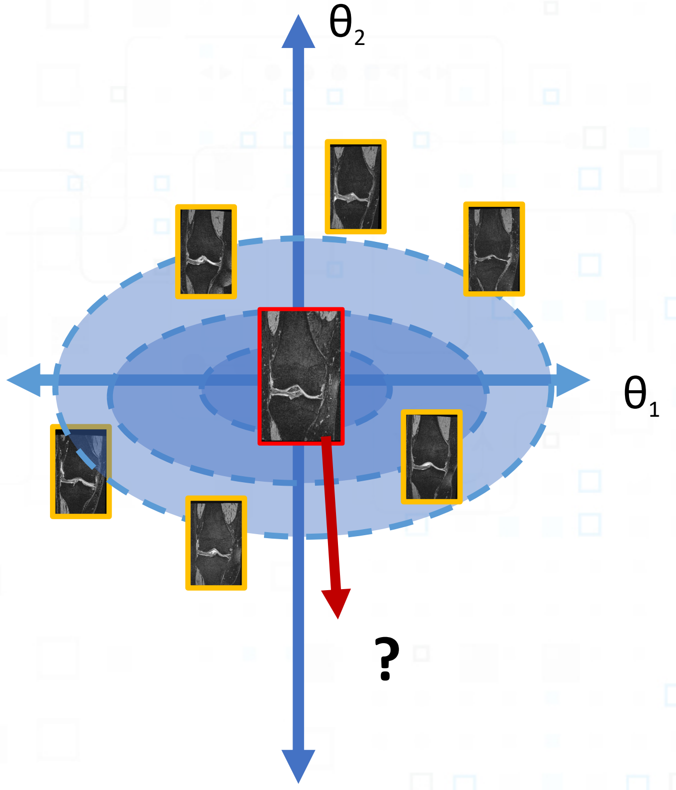
1st problem: How far should we move the source image?
2nd problem: Which direction(s) are useful for augmentation?



Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.

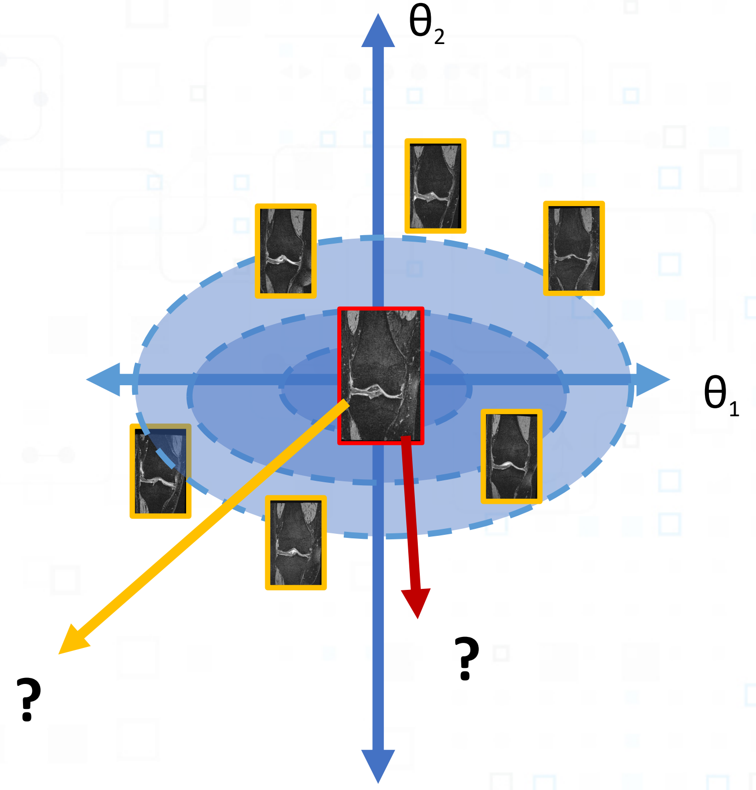
1st problem: How far should we move the source image?
2nd problem: Which direction(s) are useful for augmentation?



Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.

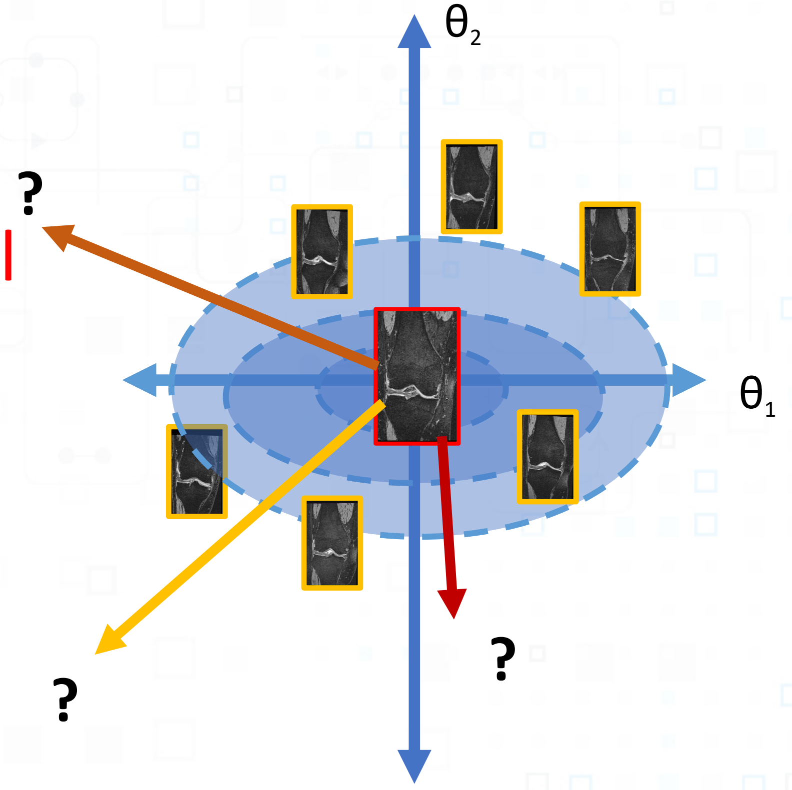
1st problem: How far should we move the source image?
2nd problem: Which direction(s) are useful for augmentation?



Statistical deformation model

Our PCA model defines an **image-specific** statistical deformation space.

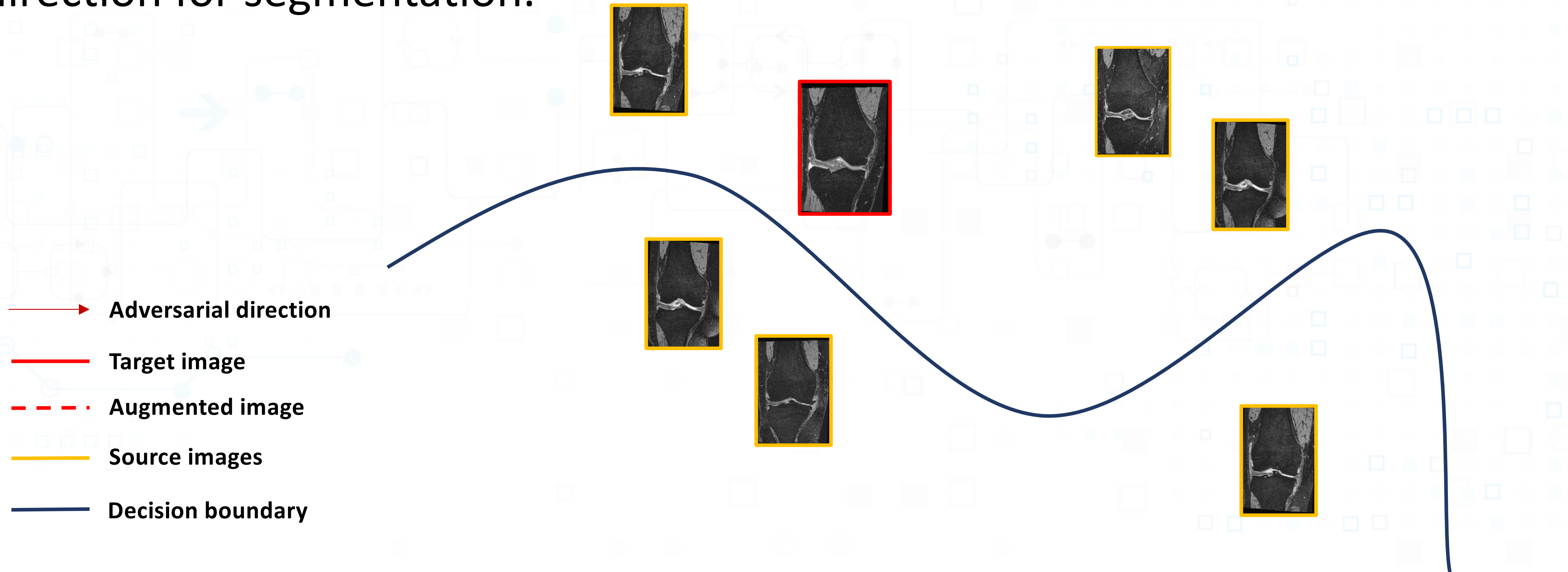
1st problem: How far should we move the source image?
2nd problem: Which direction(s) are useful for augmentation?



Adversarial direction

Problem: Which direction(s) are useful for augmentation?

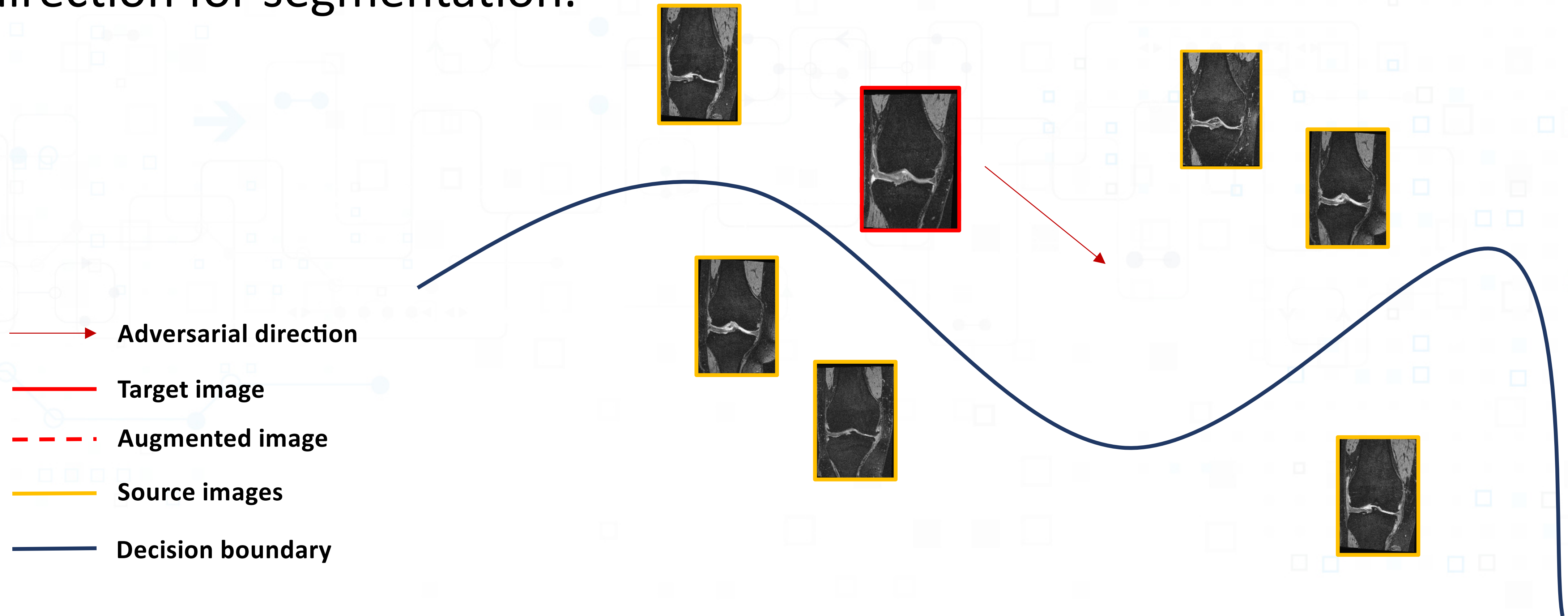
Solution: We use a segmentation network to obtain an adversarial direction for segmentation.



Adversarial direction

Problem: Which direction(s) are useful for augmentation?

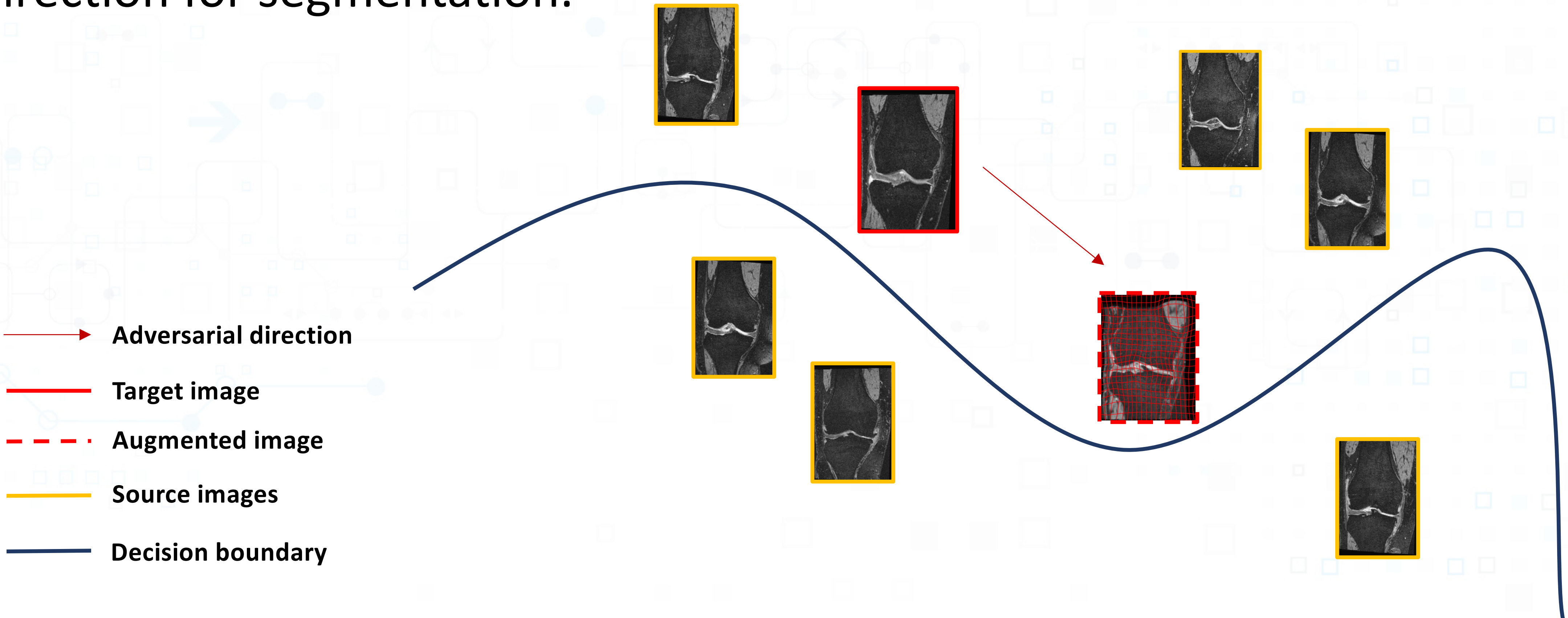
Solution: We use a segmentation network to obtain an adversarial direction for segmentation.



Adversarial direction

Problem: Which direction(s) are useful for augmentation?

Solution: We use a segmentation network to obtain an adversarial direction for segmentation.



Adversarial update

3) Using the PCA space, we update the parameters in the adversarial direction.

$$\theta = \frac{fr}{\sqrt{\sum_{i=1}^{B} f_i^2 / \lambda_i}}$$

Adversarial update

3) Using the PCA space, we update the parameters in the adversarial direction.

Adversarial deformation direction

$$\theta = \frac{fr}{\sqrt{\sum_{i=1}^{B} f_i^2/\lambda_i}}$$

Adversarial update

3) Using the PCA space, we update the parameters in the adversarial direction.

Adversarial deformation direction

How far (in std.) we are moving out from the mean

$$\theta = \frac{fr}{\sqrt{\sum_{i=1}^{B} f_i^2/\lambda_i}}$$

Adversarial update

3) Using the PCA space, we update the parameters in the adversarial direction.

Adversarial deformation direction

How far (in std.) we are moving out from the mean

$$\theta = \frac{fr}{\sqrt{\sum_{i=1}^{B} f_i^2/\lambda_i}}$$

Normalization

Adversarial update

3) Using the PCA space, we update the parameters in the adversarial direction.

Adversarial deformation direction

How far (in std.) we are moving out from the mean

$$\theta = \frac{fr}{\sqrt{\sum_{i=1}^{B} f_i^2/\lambda_i}}$$

Normalization

Adversarial update

3) Using the PCA space, we update the parameters in the adversarial direction.

Adversarial deformation direction

How far (in std.) we are moving out from the mean

$$\theta = \frac{fr}{\sqrt{\sum_{i=1}^{B} f_i^2/\lambda_i}}$$

Variance of distribution

Normalization

Experiments

For illustration we compare the following approaches:

(see paper for more experimental results)

Method	Adversarial Direction	PCA Model	Step size	Registration Method
AdvAffine	✅	❌	Fixed	Affine
$\text{AdvEigAug}_{r=2}$			Fixed	Disp.
$\text{AdvEigAug}_{rand\_r}$			Random	Disp.

Experiments

For illustration we compare the following approaches:
(see paper for more experimental results)

Method	4 training samples Dice (%)	8 training samples Dice (%)
NoAug	75.2	77.1
Upper bound (n=200)	82.5	

We use knee MR images from the OAI dataset[1] in our experiments.

[1] https://nda.nih.gov/oai

Experiments

For illustration we compare the following approaches:
(see paper for more experimental results)

Method	4 training samples Dice (%)	8 training samples Dice (%)
NoAug	75.2	77.1
Brainstorm	77.7	79.4
Upper bound (n=200)	82.5	

Experiments

For illustration we compare the following approaches:
(see paper for more experimental results)

Method	4 training samples Dice (%)	8 training samples Dice (%)
NoAug	75.2	77.1
Brainstorm	77.7	79.4
AdvAffine	75.1	77.8
Upper bound (n=200)	82.5	

Experiments

For illustration we compare the following approaches:
(see paper for more experimental results)

Method	4 training samples Dice (%)	8 training samples Dice (%)
NoAug	75.2	77.1
Brainstorm	77.7	79.4
AdvAffine	75.1	77.8
AdvEigAug_$_{r=2}$	76.1	78.7
Upper bound (n=200)	82.5	

ECCV'20 ONLINE
23-28 AUGUST 2020

Experiments

For illustration we compare the following approaches:
(see paper for more experimental results)

Method	4 training samples Dice (%)	8 training samples Dice (%)
NoAug	75.2	77.1
Brainstorm	77.7	79.4
AdvAffine	75.1	77.8
AdvEigAug_$_{r=2}$	76.1	78.7
AdvEigAug_$_{rand_r}$	**78.4**	**81.0**
Upper bound (n=200)	82.5	

Summary

- We efficiently estimate statistical deformation models with a deep registration network and PCA to obtain realistic deformations.
- We integrate our statistical deformation models into an adversarial data augmentation strategy.

Doing so, we can train deep segmentation models with few images.