DLMI 2025

Semi/weakly-supervised Learning for Medical Image Segmentation

Christian Desrosiers

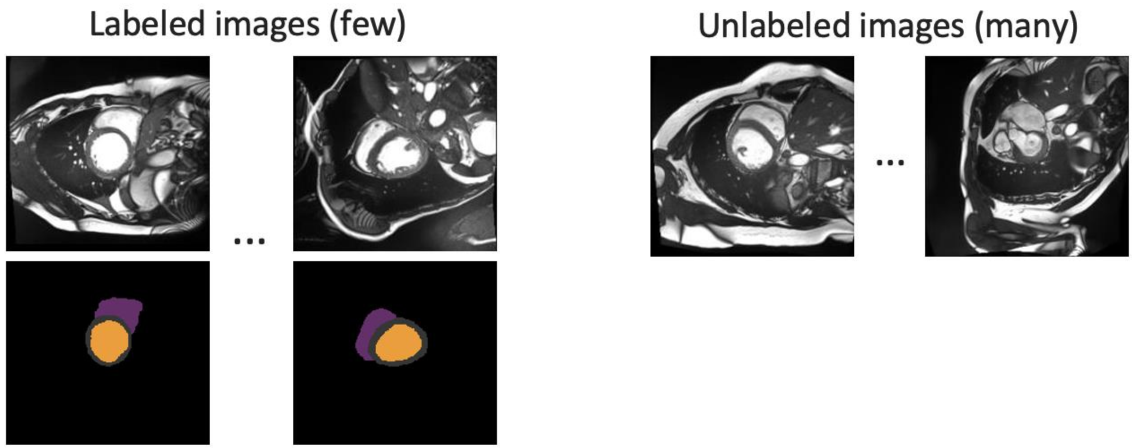
ÉTS Montreal, Canada

ÉTS Le génie pour l'industrie

Outline

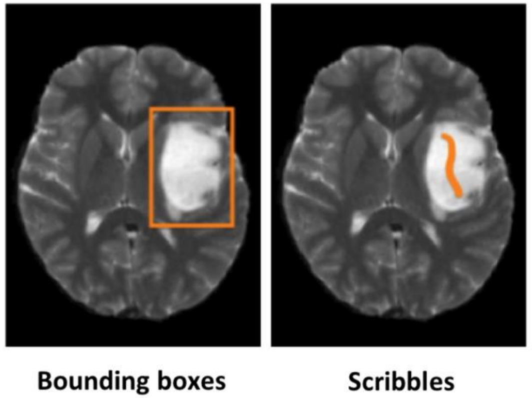
Part I: Semi-supervised learning

1. Introduction
2. Adversarial learning
3. Pseudo-labeling & entropy min.
4. Consistency regularization
5. Self-supervised learning



Part II: Weakly-supervised learning

1. Introduction
2. Pixel-level annotations
3. Image-level annotations



Part I:

Semi-supervised learning

Importance of unlabeled data

Training deep neural nets requires **lots** of labeled data

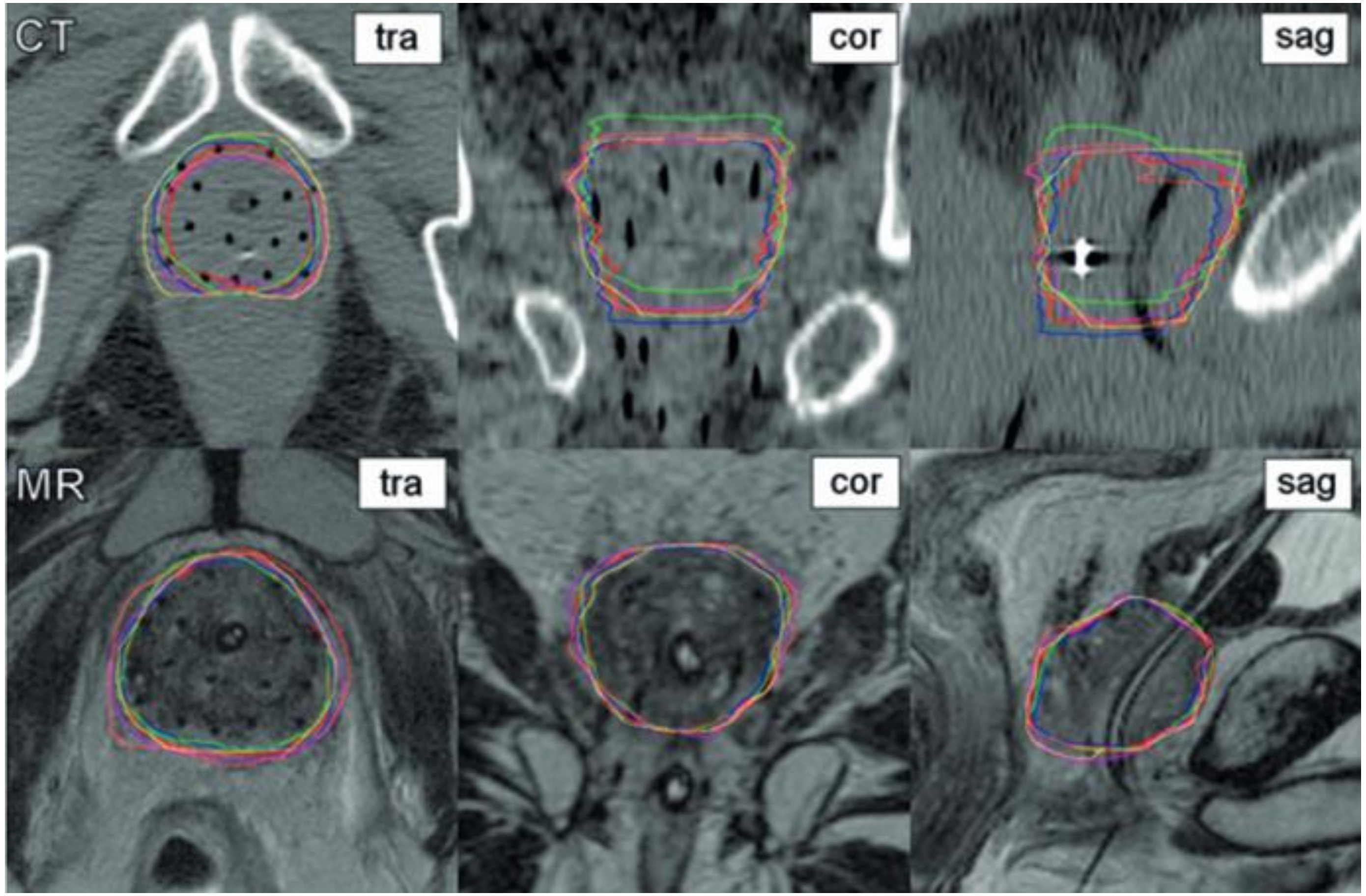
Source: https://cs.stanford.edu/people/karpathy/cnnembed/

ImageNet

- Over 14M annotated images
- More than 20,000 classes

Importance of unlabeled data

However, annotating data can be **hard** and **expensive** in some applications...



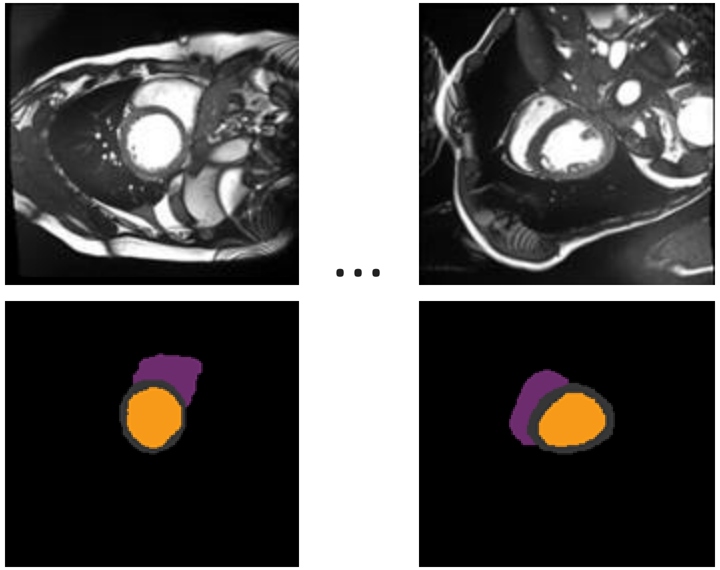
Challenges:

- Volumetric images
- Low contrast regions
- Can only be done by trained experts

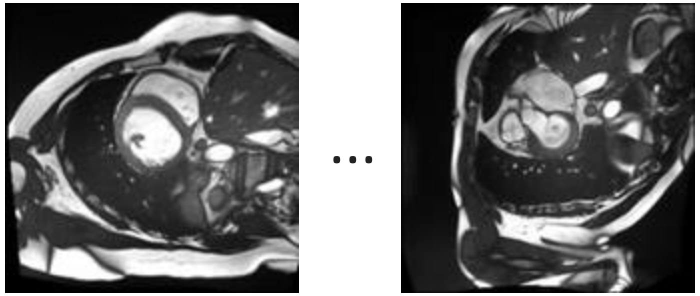
... but unlabeled data is often available for **free**

Learning with unlabeled images

Labeled images (few)

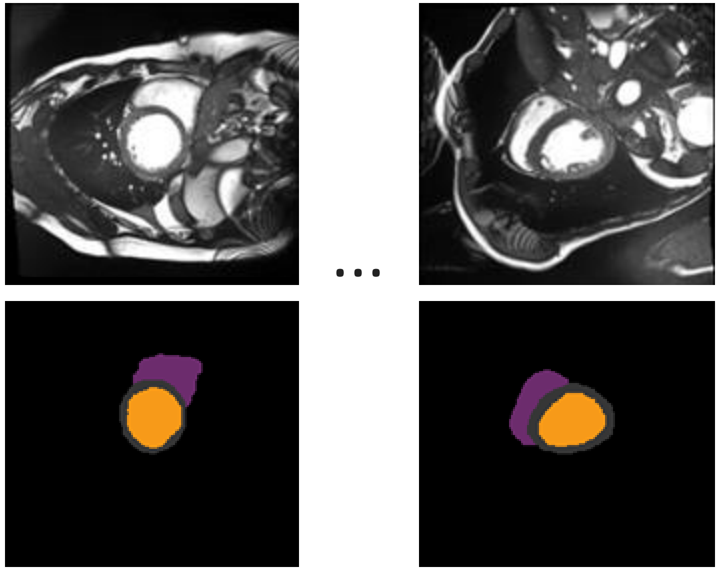


Unlabeled images (many)

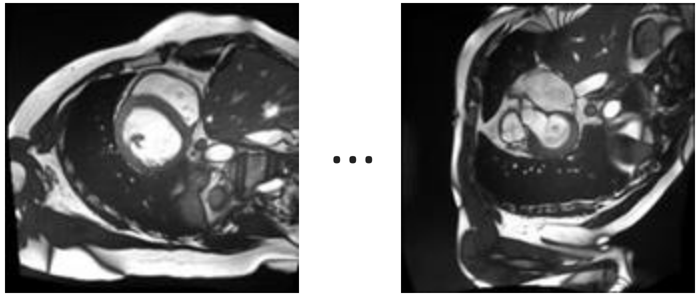


Learning with unlabeled images

Labeled images (few)



Unlabeled images (many)



How can we use this information
to learn segmentation ?

General formulation

Labeled images (few)

$$\mathcal{D}_s = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{N_s}$$

Unlabeled images (many)

$$\mathcal{D}_u = \{\mathbf{X}^{(u)}\}_{u=1}^{N_u}$$

where:

$\mathbf{X}^{(i)}, \mathbf{X}^{(u)} \in \mathbb{R}^{|\Omega|}$ are images on the set of pixels $\boldsymbol{\Omega}$

$\mathbf{Y}^{(i)} \in \{0, 1\}^{|\Omega| \times |\mathcal{C}|}$ are ground-truth mask for classes $\mathcal{C}$

General formulation

Labeled images (few)

$$\mathcal{D}_s = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{N_s}$$

Unlabeled images (many)

$$\mathcal{D}_u = \{\mathbf{X}^{(u)}\}_{u=1}^{N_u}$$

where:

$\mathbf{X}^{(i)}, \mathbf{X}^{(u)} \in \mathbb{R}^{|\Omega|}$ are images on the set of pixels $\mathbf{\Omega}$

$\mathbf{Y}^{(i)} \in \{0, 1\}^{|\Omega| \times |\mathcal{C}|}$ are ground-truth mask for classes $\mathcal{C}$

Objective:

$$\mathcal{L}_{tot}(\theta) = \underbrace{\frac{1}{N_s} \sum_{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}) \in \mathcal{D}_s} \ell_s\big(\mathbf{S}_\theta^{(i)}, \mathbf{Y}^{(i)}\big)}_{\text{Supervised loss } \mathcal{L}_s} + \underbrace{\frac{\lambda}{N_u} \sum_{\mathbf{X}^{(u)} \in \mathcal{D}_u} \ell_u\big(\mathbf{S}_\theta^{(u)}\big)}_{\text{Unsupervised loss } \mathcal{L}_u}$$

General formulation

Labeled images (few)

$$\mathcal{D}_s = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{N_s}$$

Unlabeled images (many)

$$\mathcal{D}_u = \{\mathbf{X}^{(u)}\}_{u=1}^{N_u}$$

where:

$\mathbf{X}^{(i)}, \mathbf{X}^{(u)} \in \mathbb{R}^{|\Omega|}$ are images on the set of pixels Ω

$\mathbf{Y}^{(i)} \in \{0, 1\}^{|\Omega| \times |\mathcal{C}|}$ are ground-truth mask for classes $\mathcal{C}$

Objective:

Trade-off weight (hyper-parameter)

$$\mathcal{L}_{tot}(\theta) = \underbrace{\frac{1}{N_s} \sum_{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}) \in \mathcal{D}_s} \ell_s(\mathbf{S}_\theta^{(i)}, \mathbf{Y}^{(i)})}_{\text{Supervised loss } \mathcal{L}_s} + \underbrace{\frac{\boxed{\lambda}}{N_u} \sum_{\mathbf{X}^{(u)} \in \mathcal{D}_u} \ell_u(\mathbf{S}_\theta^{(u)})}_{\text{Unsupervised loss } \mathcal{L}_u}$$

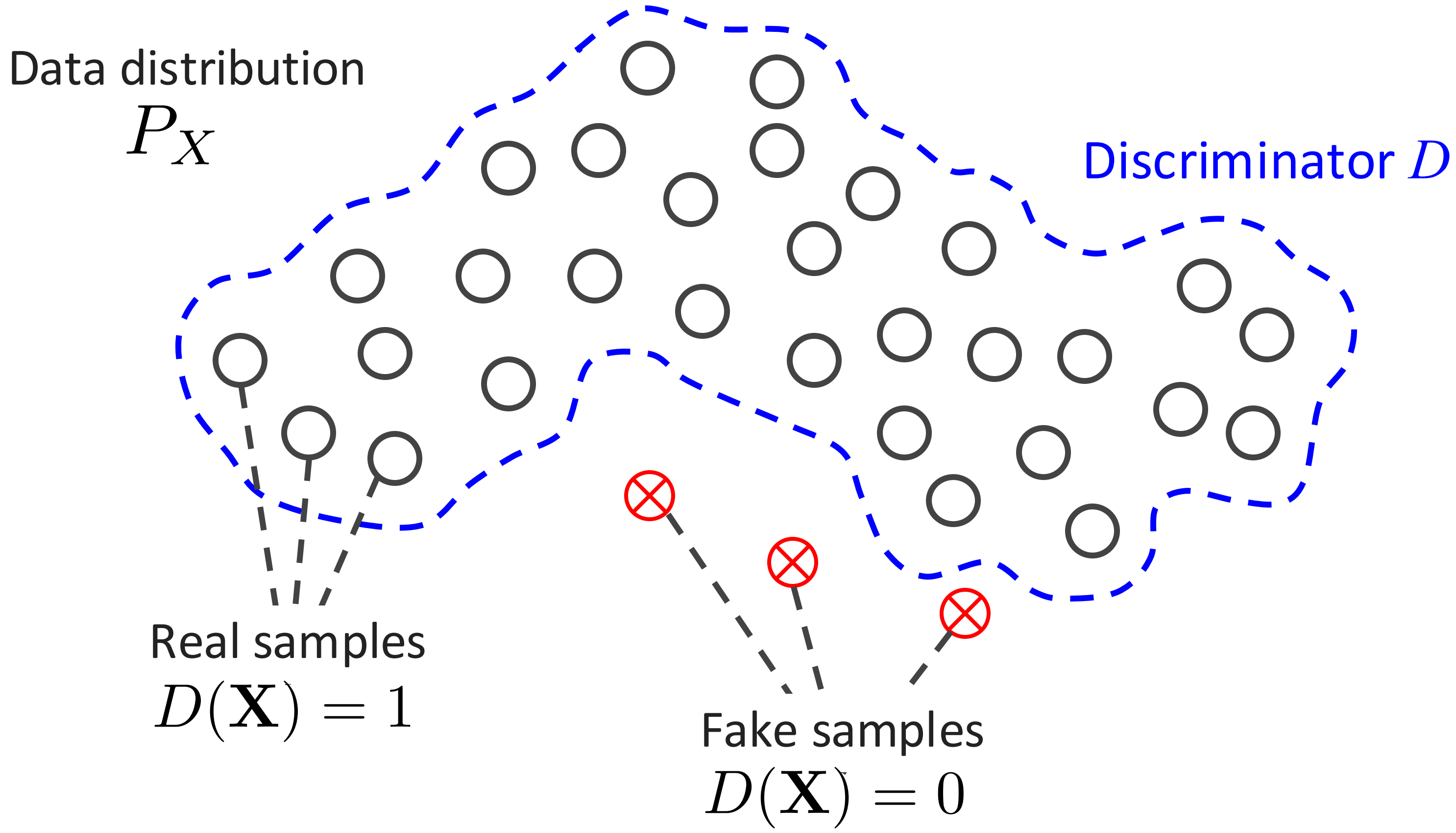
Adversarial learning

for semi-supervised segmentation

Adversarial learning

Basic idea:

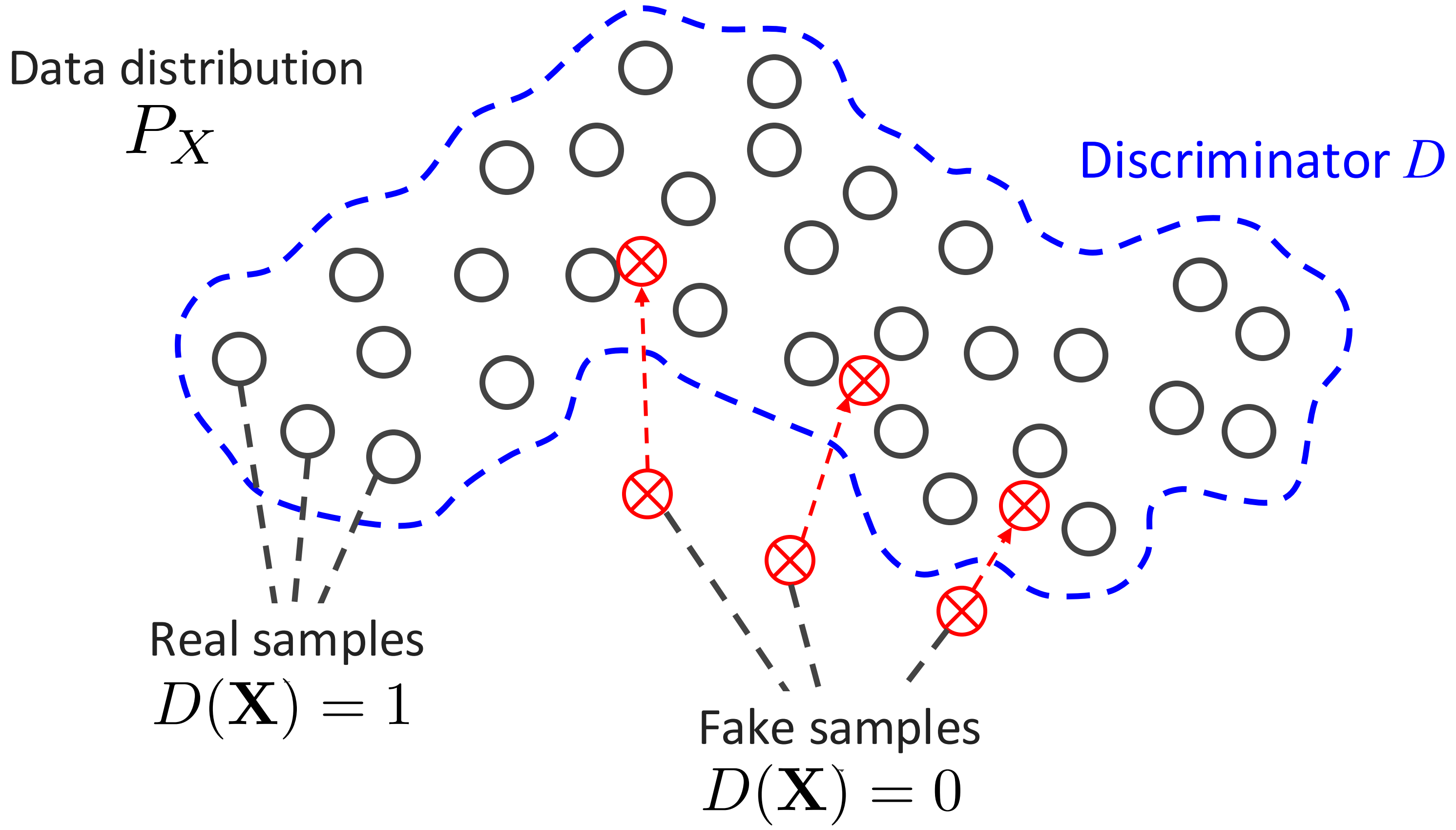
Learn the data distribution using a classifier (the discriminator)



Adversarial learning

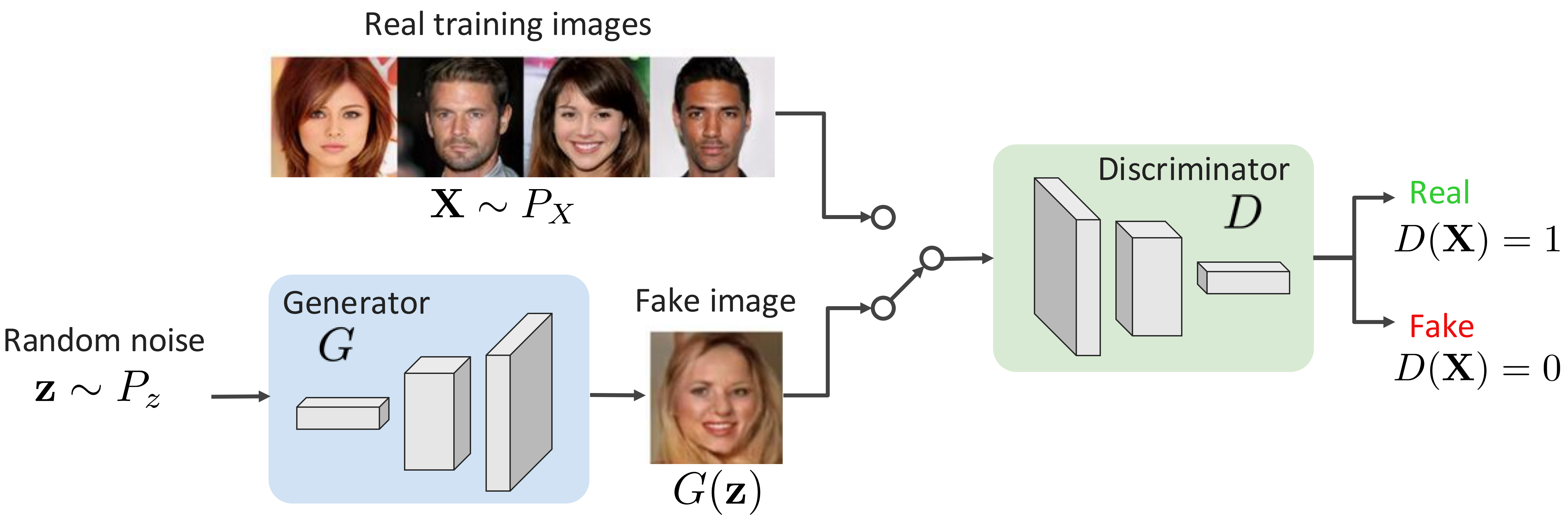
Basic idea:

Learn the data distribution using a classifier (the discriminator)

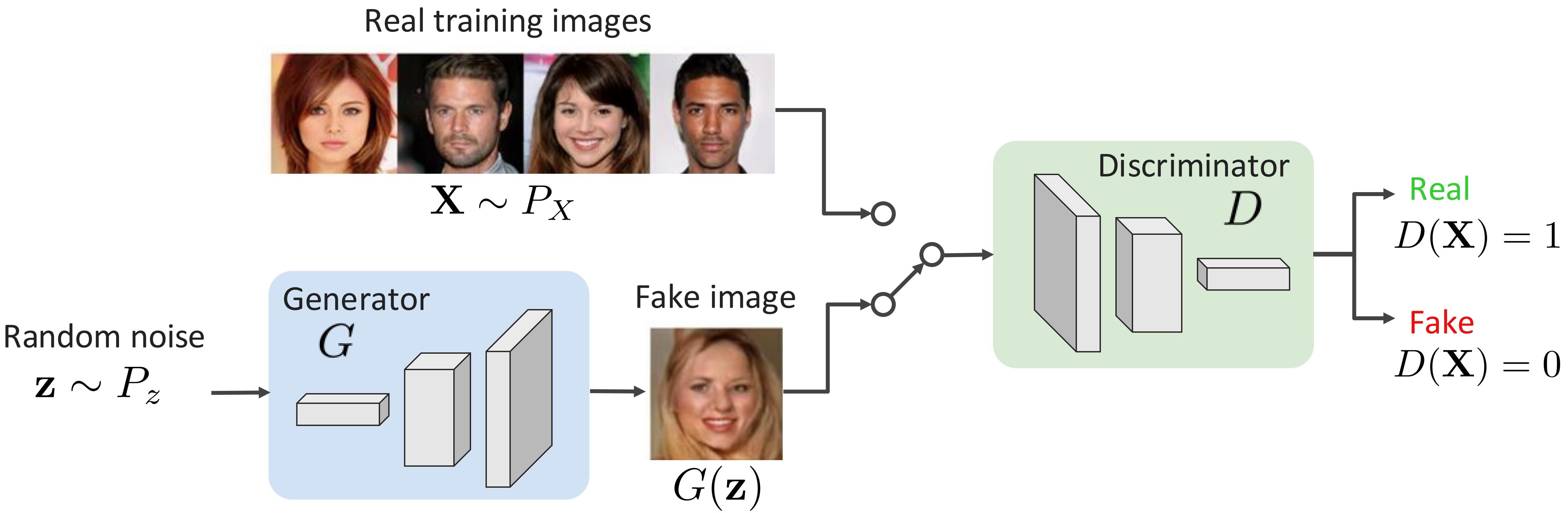


Objective: Generate samples in the distribution of real data

Generative adversarial network (GAN)

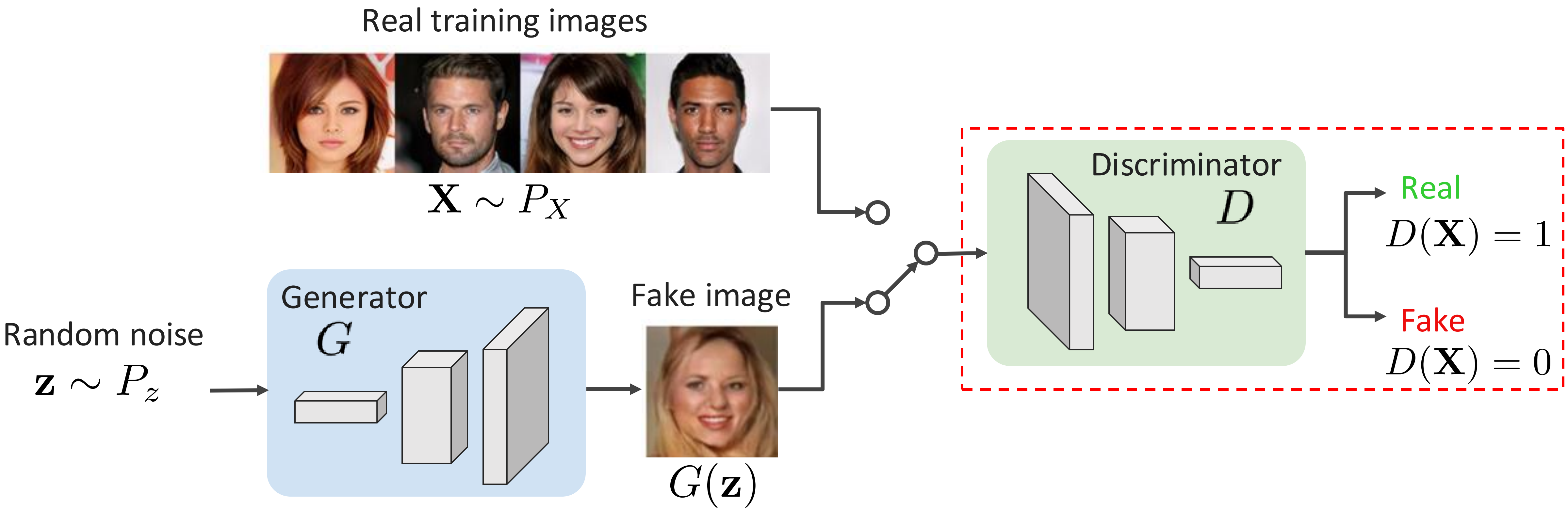


Generative adversarial network (GAN)



How to make sure that generated images look real ?

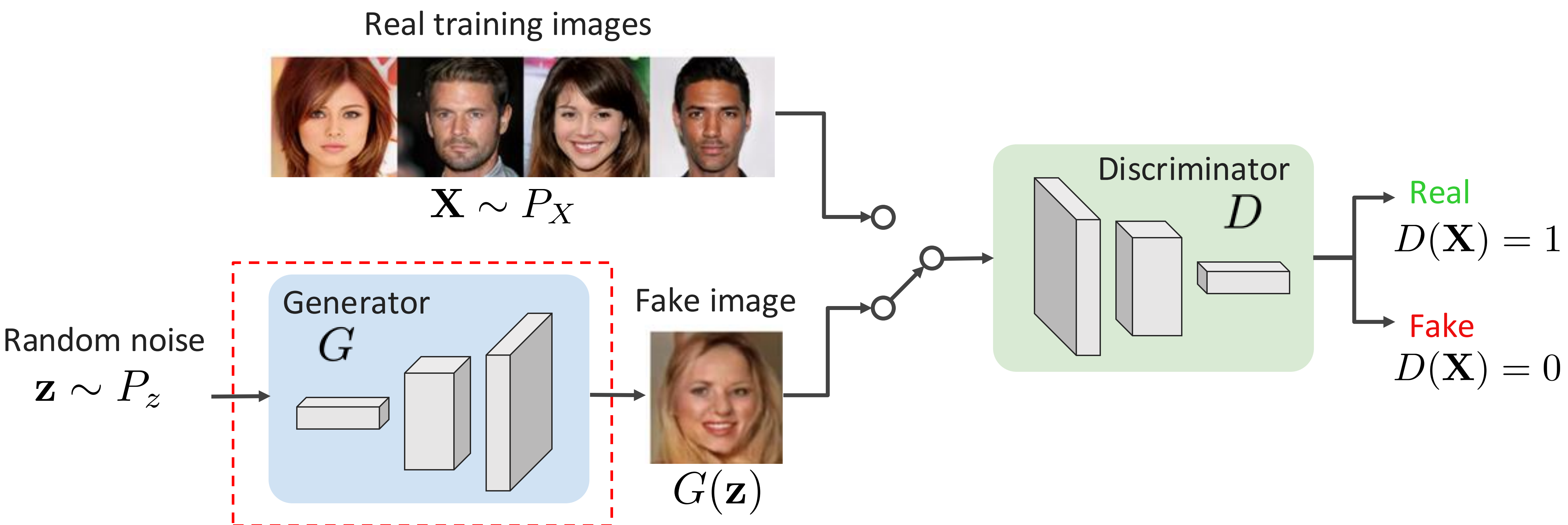
Generative adversarial network (GAN)



Training the discriminator (cross-entropy):

$$\min_{D} \underbrace{\mathbb{E}_{\mathbf{X}\sim P_X}\big[-\log D(\mathbf{X})\big]}_{\text{Output '1' for real images}} + \underbrace{\mathbb{E}_{\mathbf{z}\sim P_z}\big[-\log(1 - D(G(\mathbf{z})))\big]}_{\text{Output '0' for generated images}}$$

Generative adversarial network (GAN)

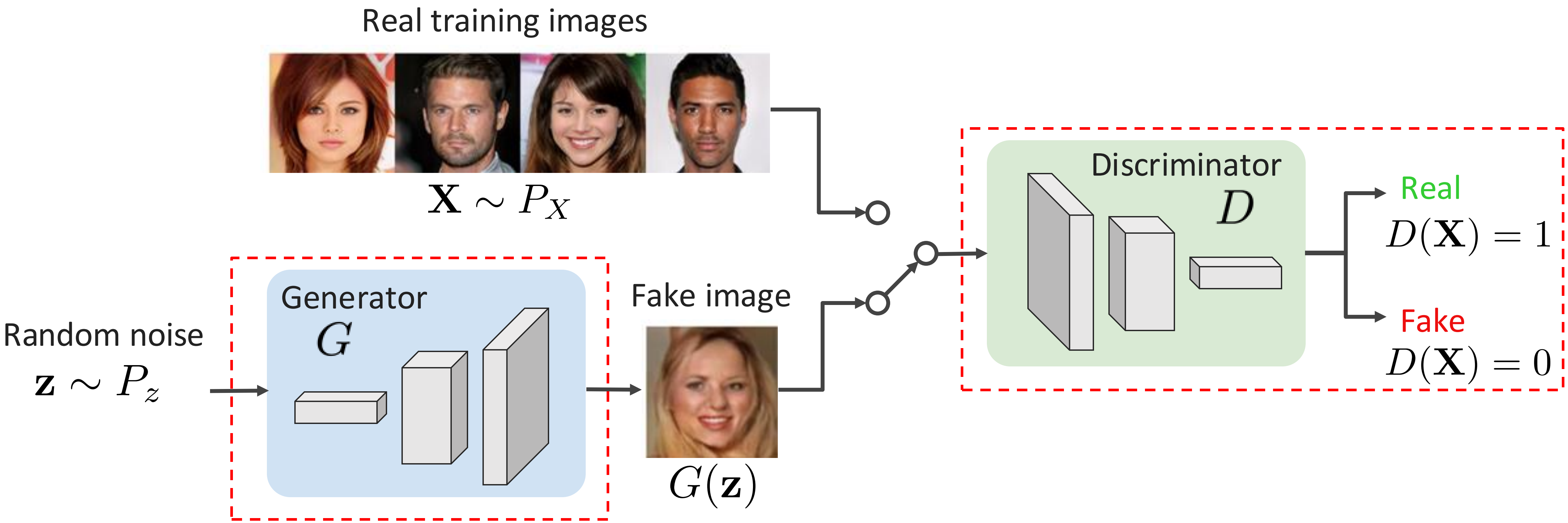


Training the generator:

$$\max_{G} \mathbb{E}_{\mathbf{z}\sim P_z}\left[-\log(1 - D(G(\mathbf{z}))\right]$$

Fool the discriminator into predicting ‘1’ for fake images

Generative adversarial network (GAN)



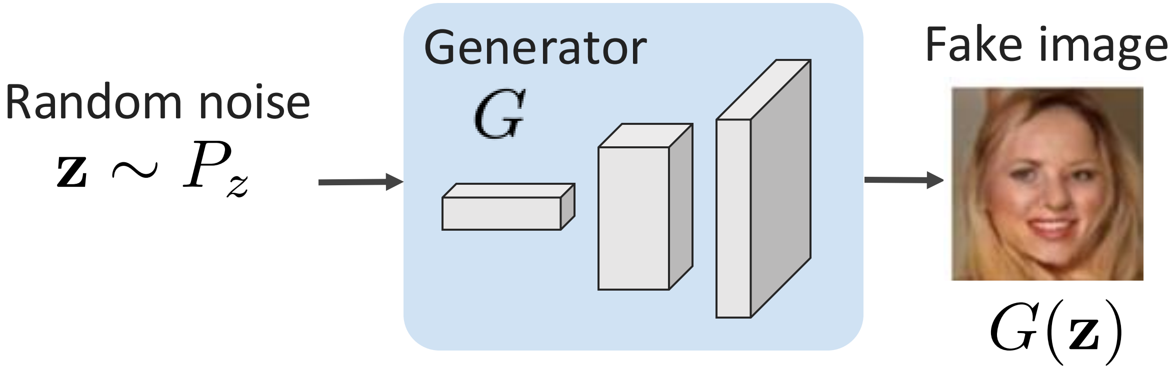
Training the whole architecture:

$$\min_G \max_D \; \mathbb{E}_{\mathbf{X} \sim P_X} \left[\log D(\mathbf{X}) \right] + \mathbb{E}_{\mathbf{z} \sim P_z} \left[\log(1 - D(G(\mathbf{z}))) \right]$$

Corresponds to a minimax problem (*more on this later...*)

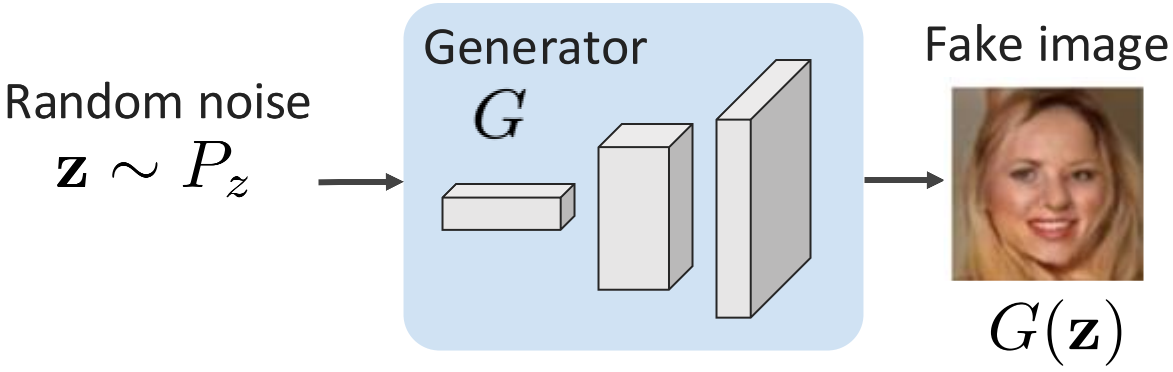
GANs for segmentation

GAN for image generation:

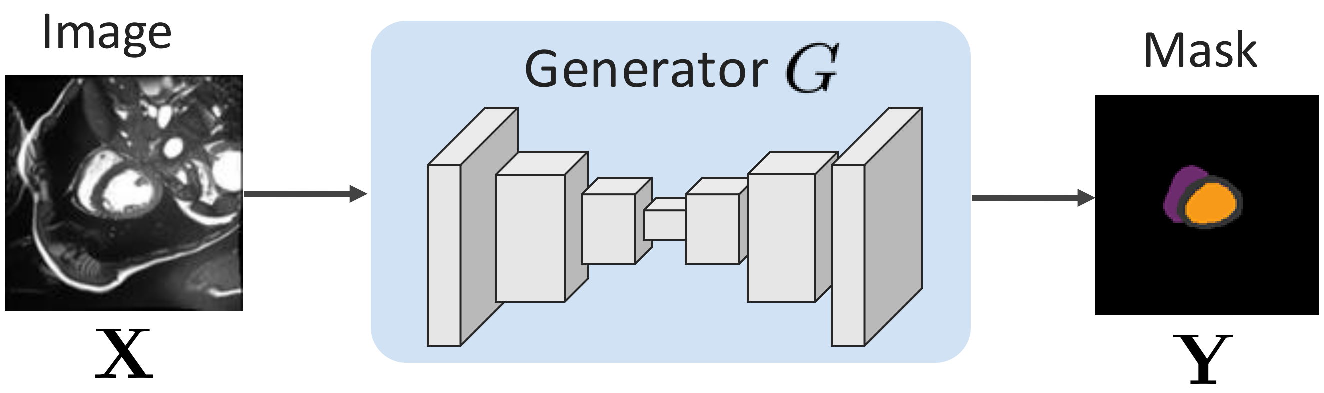


GANs for segmentation

GAN for image generation:

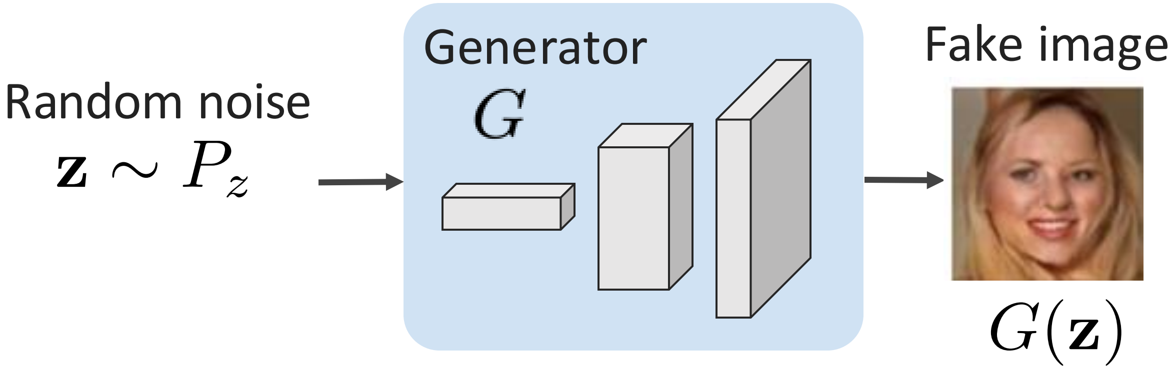


GAN for image segmentation:

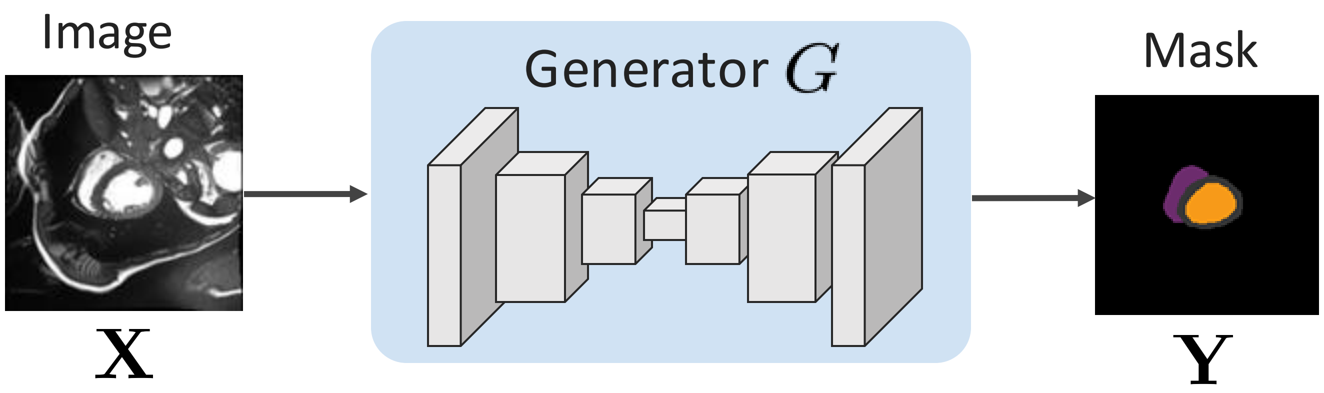


GANs for segmentation

GAN for image generation:



GAN for image segmentation:



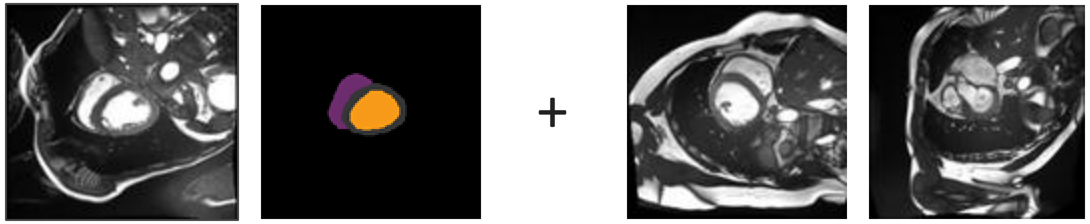
We are now modeling the distribution of *segmentation masks*

The generator is a segmentation network (encoder-decoder)

Adversarial semi-supervised segmentation

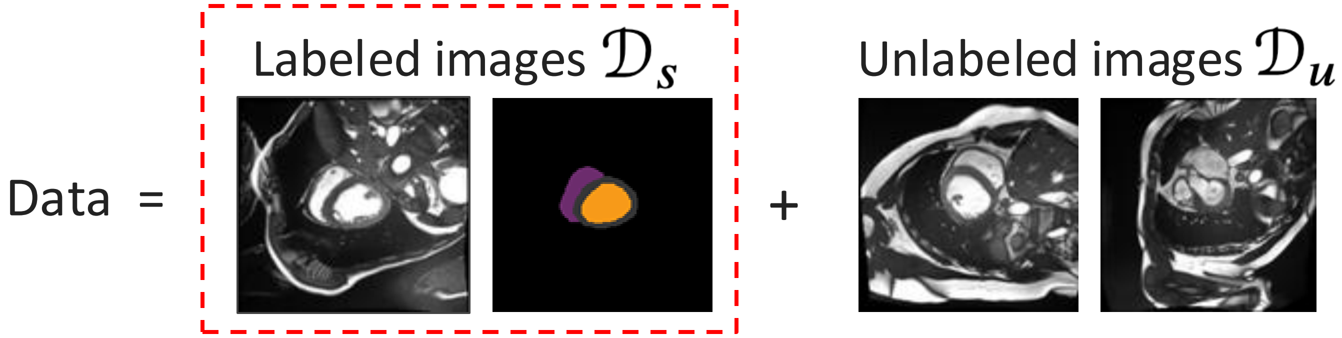
Basic idea: Learn to generate segmentation masks which can't be differentiated from ground-truth (GT)

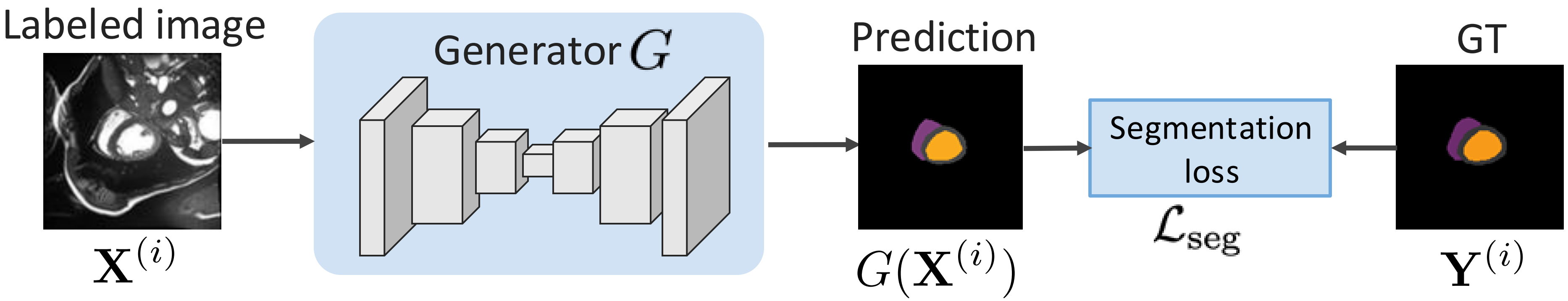
Labeled images $\mathcal{D}_s$ Unlabeled images $\mathcal{D}_u$

Data = 

Adversarial semi-supervised segmentation

Basic idea: Learn to generate segmentation masks which can't be differentiated from ground-truth (GT)



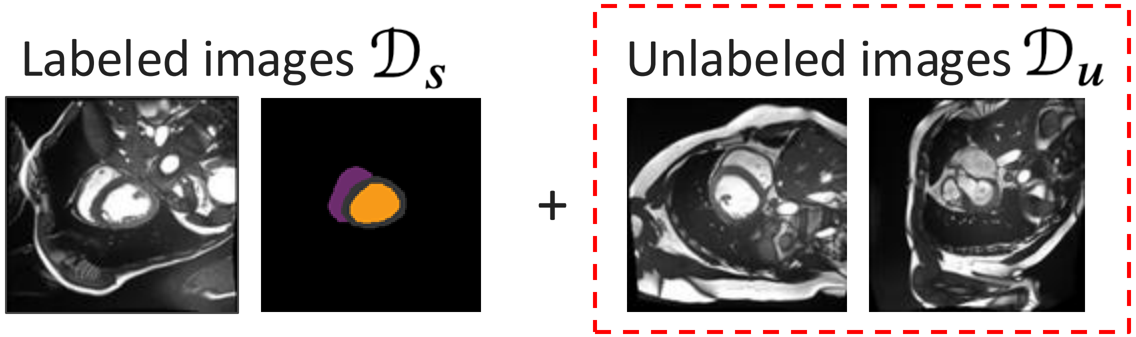


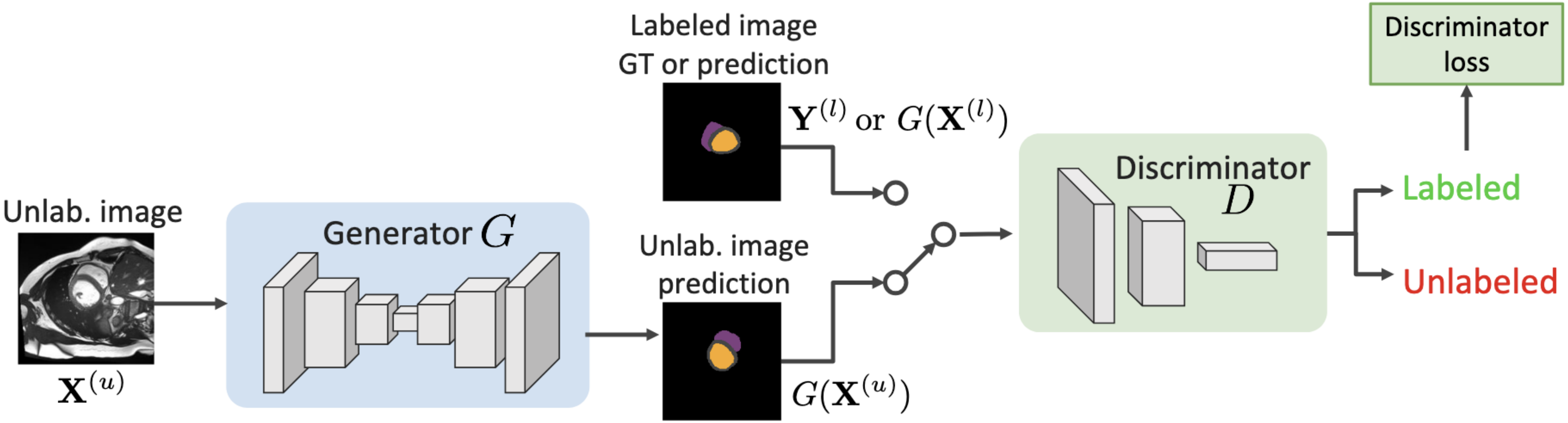
$$\mathcal{L}_{sup}(G) = \frac{1}{N_s} \sum_{(\mathbf{X}^{(i)},\mathbf{Y}^{(i)}) \in \mathcal{D}_s} \ell_s(G(\mathbf{X}^{(i)}), \mathbf{Y}^{(i)})$$

Adversarial semi-supervised segmentation

Basic idea: Learn to generate segmentation masks which can't be differentiated from ground-truth (GT)

Data =



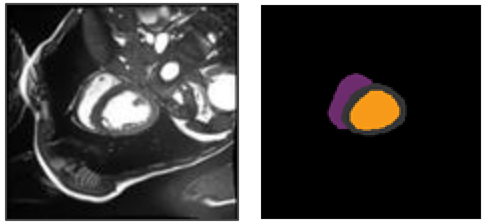
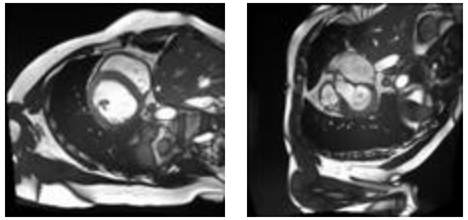


$$\mathcal{L}_{dis}(G,D) = \frac{1}{N_s}\sum_{\mathbf{X}^{(i)}\in\mathcal{D}_s} \ell_{dis}\big(D(G(\mathbf{X}^{(i)})),1\big) + \frac{1}{N_u}\sum_{\mathbf{X}^{(u)}\in\mathcal{D}_u}\ell_{dis}\big(D(G(\mathbf{X}^{(u)})),0\big)$$

Adversarial semi-supervised segmentation

Basic idea: Learn to generate segmentation masks which can't be differentiated from ground-truth (GT)

Labeled images $\mathcal{D}_s$ Unlabeled images $\mathcal{D}_u$

Data =  + 

Both labeled and unlabeled:

Controls the trade-off between the two losses

$$\min_G \max_D \mathcal{L}_{adv}(G,D) = \underbrace{\mathcal{L}_{sup}(G)}_{\text{Supervised loss}} - \underbrace{\lambda \mathcal{L}_{dis}(G,D)}_{\text{Discriminator loss}}$$

Adversarial semi-supervised segmentation

Adversarial network for semi-supervised segmentation of histological images

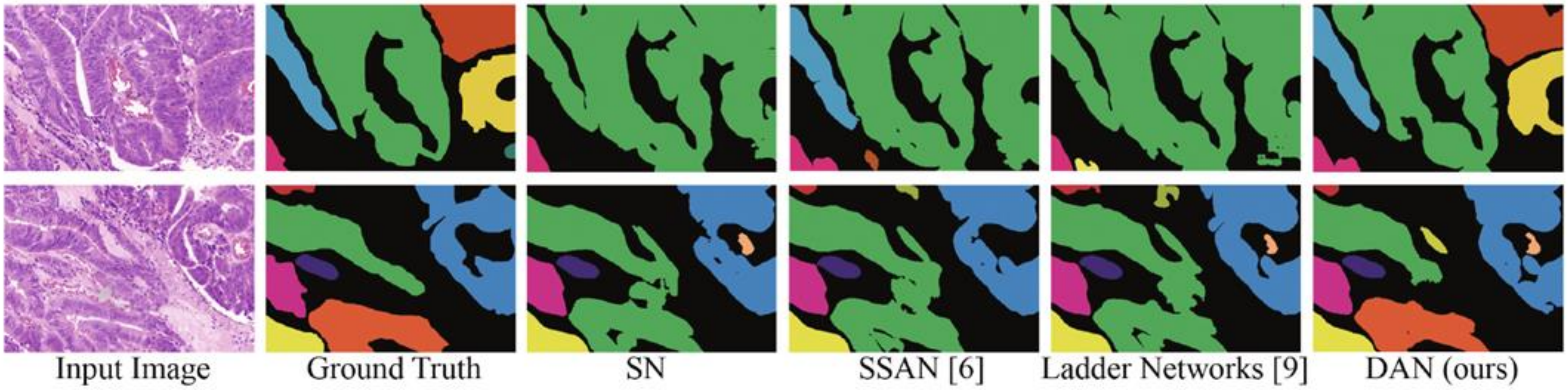
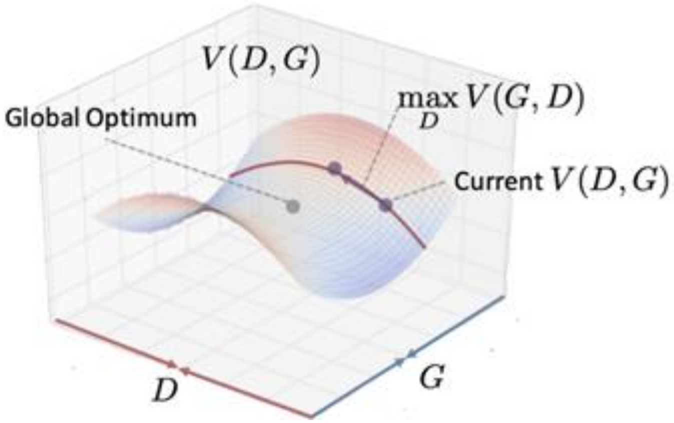
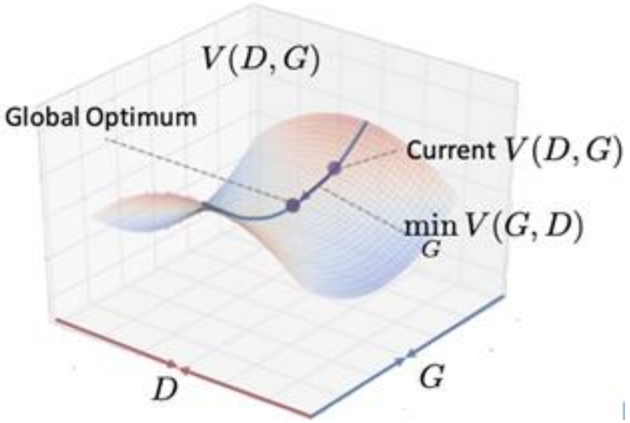


Image from Zhang, Y., et al. "Deep adversarial networks for biomedical image segmentation utilizing unannotated images." *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*. 2017.

Challenges of adversarial learning

1) Unstable optimization of minimax problem

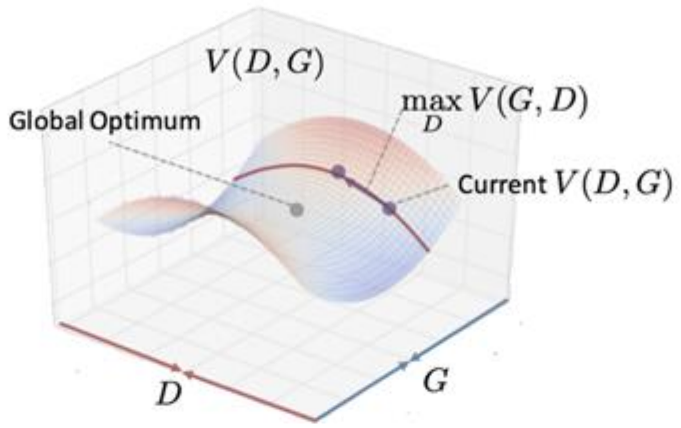


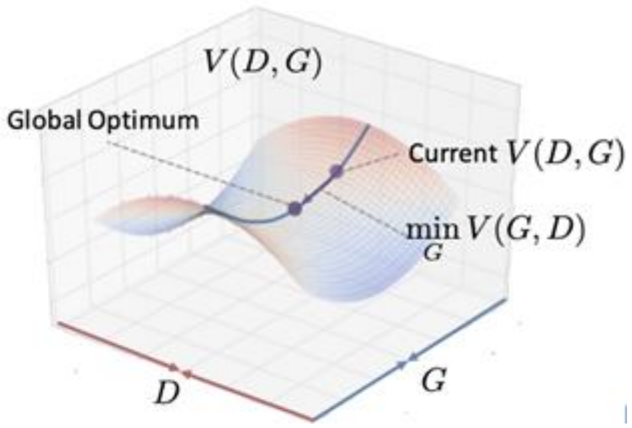


Ghosh, A., et al. "Multi-agent diverse generative adversarial networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
Chang, Mark. “Generative Adversarial Networks”, published online, 2016

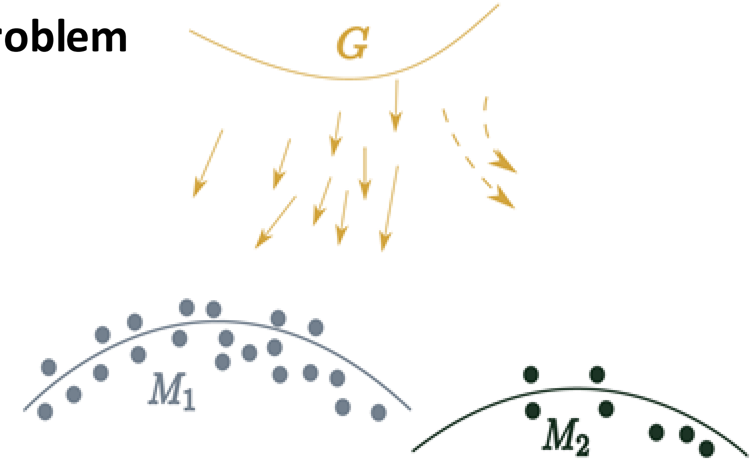
Challenges of adversarial learning

1) Unstable optimization of minimax problem





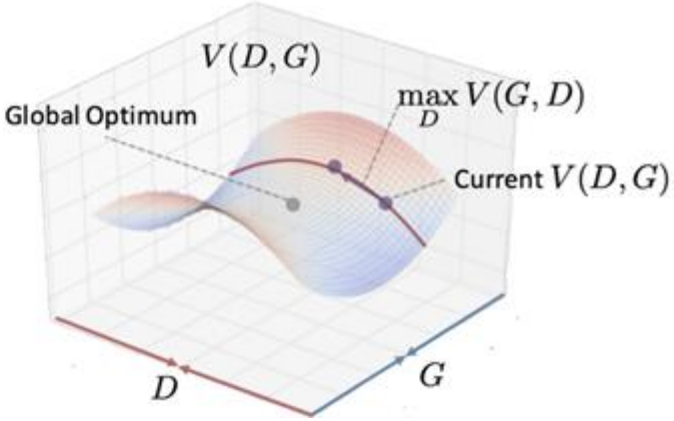
2) Mode collapse problem

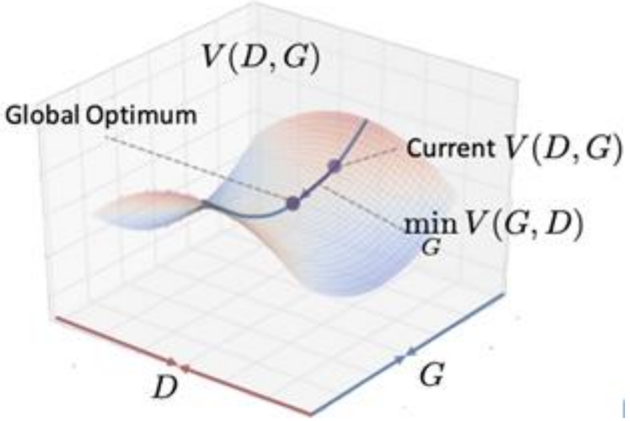


Ghosh, A., et al. "Multi-agent diverse generative adversarial networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
Chang, Mark. “Generative Adversarial Networks”, published online, 2016

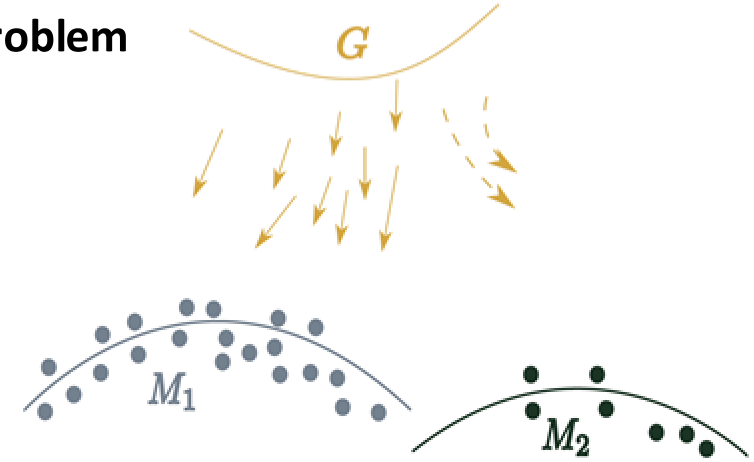
Challenges of adversarial learning

1) Unstable optimization of minimax problem





2) Mode collapse problem



Various solutions:

- Spectral normalization (Miyato *et al*., 2018)
- Wasserstein GANs (Arjovsky *et al*., 2017)
- LSGANs (Mao *et al*., 2017)
- etc.

Ghosh, A., et al. "Multi-agent diverse generative adversarial networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
Chang, Mark. "Generative Adversarial Networks", published online, 2016

Pseudo-labeling & entropy minimization

for semi-supervised segmentation

SSL methods using pseudo-labeling

Pseudo-labeling (self-training):

$$\ell_u(\mathbf{X}^{(u)}) = \frac{\sum_{p\in\Omega} \mathbb{1}\big(Conf(\mathbf{s}_p^{(u)}) \geq \tau\big) \cdot \mathcal{H}(\hat{\mathbf{y}}_p^{(u)}, \mathbf{s}_p^{(u)})}{\sum_{p\in\Omega} \mathbb{1}\big(Conf(\mathbf{s}_p^{(u)}) \geq \tau\big)}$$

$$\hat{y}_{p,k}^{(u)} = \begin{cases} 1, & \text{if } k = \arg\max_{k'} s_{p,k'}^{(u)} \\ 0, & \text{else} \end{cases}$$

Key idea:

- Convert confident predictions for unlabeled samples into pseudo-labels
- Use the pseudo-labeled samples in a standard loss (e.g., cross-entropy)
- Example of confidence score: $Conf(\mathbf{s}_p^{(u)}) = \max_k s_{p,k}^{(u)}$.

SSL methods using pseudo-labeling

Pseudo-labeling (self-training):

$$\ell_u(\mathbf{X}^{(u)}) = \frac{\sum_{p\in\Omega} \mathbb{1}\big(\underbrace{\boxed{Conf(\mathbf{s}_p^{(u)})}}_{\text{Confidence score}} \geq \tau\big)\cdot \underbrace{\boxed{\mathcal{H}(\hat{\mathbf{y}}_p^{(u)}, \mathbf{s}_p^{(u)})}}_{\text{Cross-entropy}}}{\sum_{p\in\Omega} \mathbb{1}\big(Conf(\mathbf{s}_p^{(u)}) \geq \tau\big)}$$

$$\underbrace{\boxed{\hat{y}_{p,k}^{(u)}}}_{\text{Pseudo-label}} = \begin{cases} 1, & \text{if } k = \arg\max_{k'} s_{p,k'}^{(u)} \\ 0, & \text{else} \end{cases}$$

Key idea:

- Convert confident predictions for unlabeled samples into pseudo-labels
- Use the pseudo-labeled samples in a standard loss (e.g., cross-entropy)
- Example of confidence score: $Conf(\mathbf{s}_p^{(u)}) = \max_k s_{p,k}^{(u)}$.

SSL methods using pseudo-labeling

Pseudo-labeling (self-training):

Confidence score | Cross-entropy

$$\ell_u(\mathbf{X}^{(u)}) = \frac{\sum_{p\in\Omega} \mathbb{1}\left(\boxed{Conf(\mathbf{s}_p^{(u)})} \geq \tau\right) \cdot \boxed{\mathcal{H}(\hat{\mathbf{y}}_p^{(u)}, \mathbf{s}_p^{(u)})}}{\sum_{p\in\Omega} \mathbb{1}\left(Conf(\mathbf{s}_p^{(u)}) \geq \tau\right)}$$

Pseudo-label

$$\boxed{\hat{y}_{p,k}^{(u)}} = \begin{cases} 1, & \text{if } k = \arg\max_{k'} s_{p,k'}^{(u)} \\ 0, & \text{else} \end{cases}$$

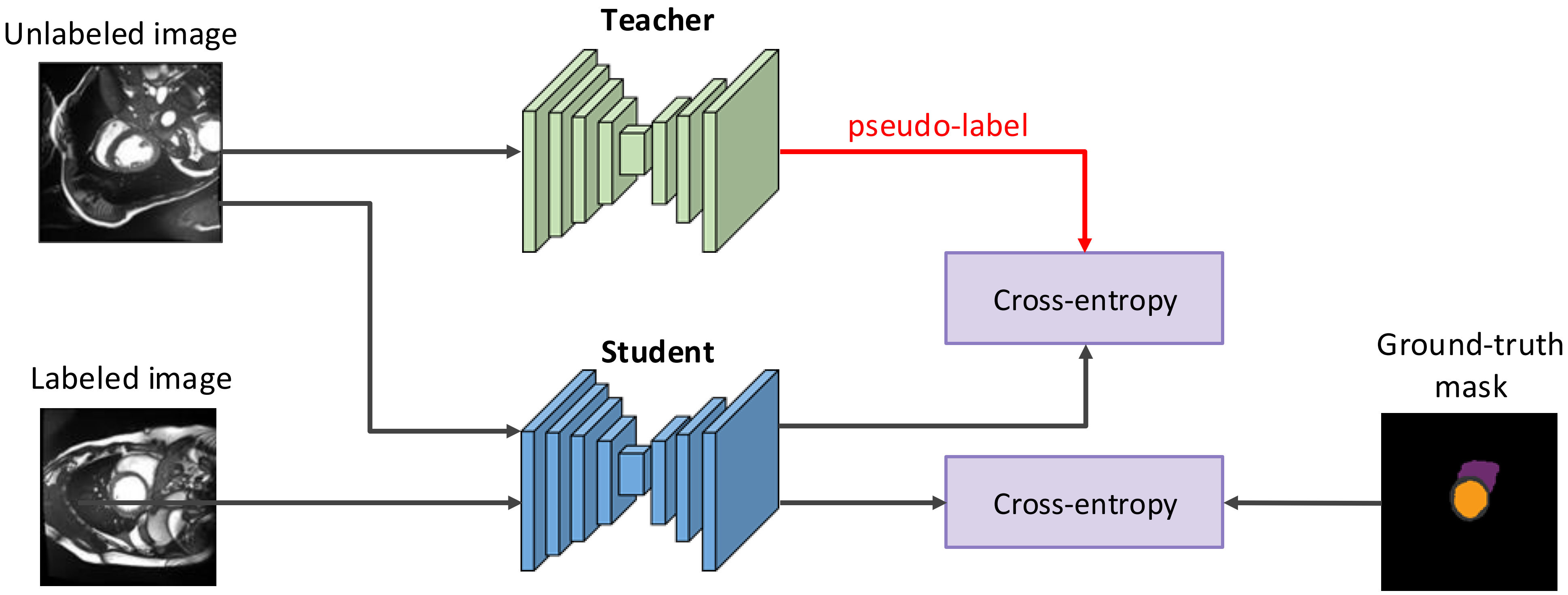
Key idea:

- Convert confident predictions for unlabeled samples into pseudo-labels
- Use the pseudo-labeled samples in a standard loss (e.g., cross-entropy)
- Example of confidence score: $Conf(\mathbf{s}_p^{(u)}) = \max_k s_{p,k}^{(u)}$.

Problem: incorrect predictions are reinforced leading to collapse

SSL methods using pseudo-labeling

Pseudo-labeling (with teacher):



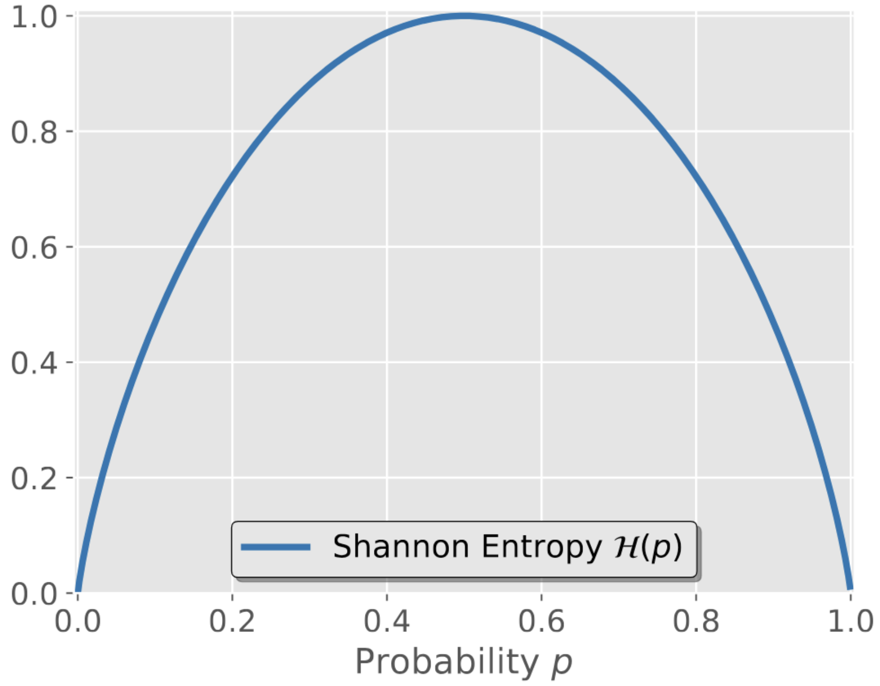
Key idea:

- Use the predictions of a **Teacher** network to generate pseudo-labels
- More stable, lower chances of collapse
- Teacher is typically pre-trained on labeled images and/or updated using EMA (*more on this later...*)

SSL methods using entropy minimization

Entropy as unsupervised loss:

$$\ell_u(\mathbf{X}^{(u)}) = -\frac{1}{|\Omega|}\sum_{p\in\Omega}\sum_{k\in\mathcal{C}} s_{p,k}^{(u)} \log s_{p,k}^{(u)}$$



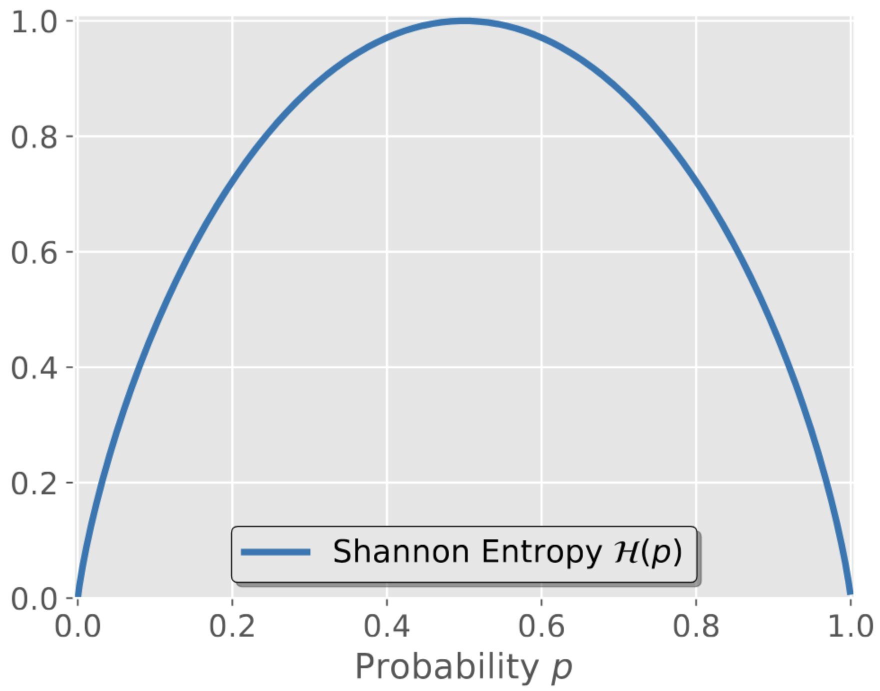
Key idea:

- Can be seen as a **<u>soft</u>** version of pseudo-labeling where all predictions are considered as confident and the pseudo-label is the prediction itself. i.e. $\hat{\mathbf{y}}_p^{(u)} = \mathbf{s}_p^{(u)}$

SSL methods using entropy minimization

Entropy as unsupervised loss:

$$\ell_u(\mathbf{X}^{(u)}) = -\frac{1}{|\Omega|}\sum_{p\in\Omega}\sum_{k\in\mathcal{C}} s_{p,k}^{(u)} \log s_{p,k}^{(u)}$$



Key idea:

- Can be seen as a **soft** version of pseudo-labeling where all predictions are considered as confident and the pseudo-label is the prediction itself. i.e. $\hat{\mathbf{y}}_p^{(u)} = \mathbf{s}_p^{(u)}$

Problem: incorrect predictions that are confident will remain « stuck »

Solution: increase the entropy loss slowly during training

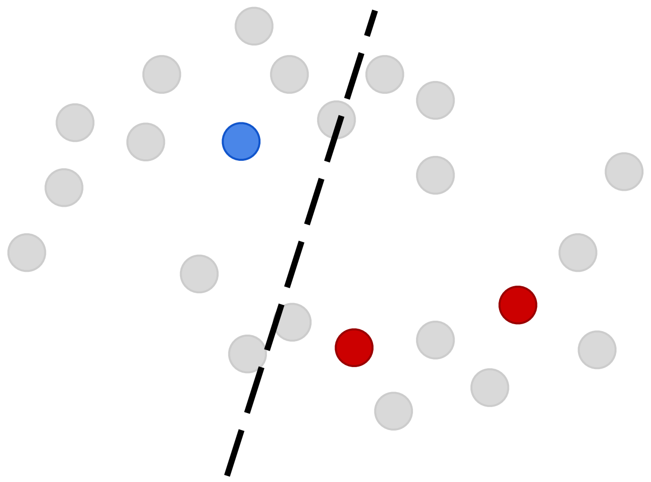
Consistency regularization

for semi-supervised segmentation

Consistency regularization for SSL

How to better use unlabeled data ?

Vanilla supervised learning

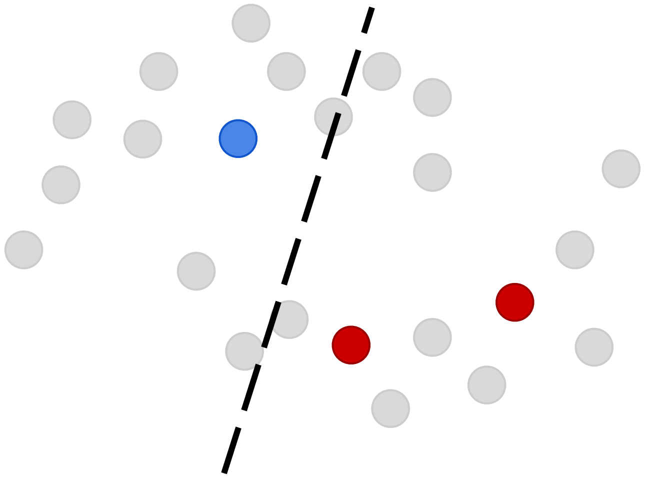


- Considers only *labeled* samples
- Overfits when few training samples

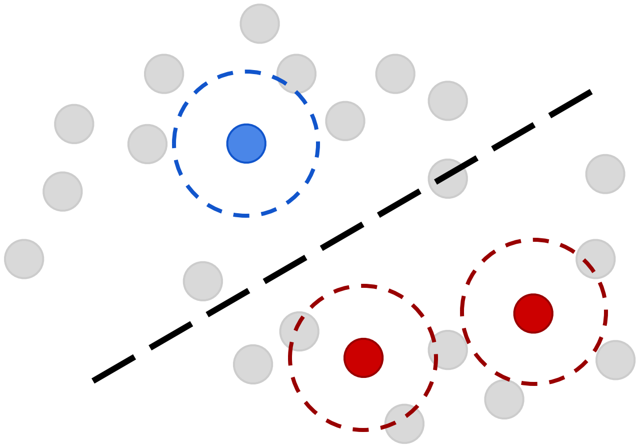
Consistency regularization for SSL

How to better use <u>unlabeled data</u> ?

Vanilla supervised learning



Data augmentation



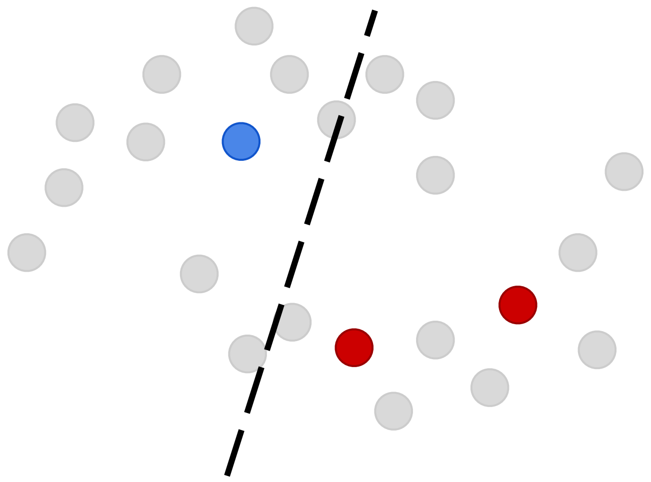
- Considers only <u>*labeled*</u> samples
- Overfits when few training samples

- Transforms <u>*labeled*</u> samples to augment the training set
- Better generalization, but not enough for semi-supervised learning

Consistency regularization for SSL

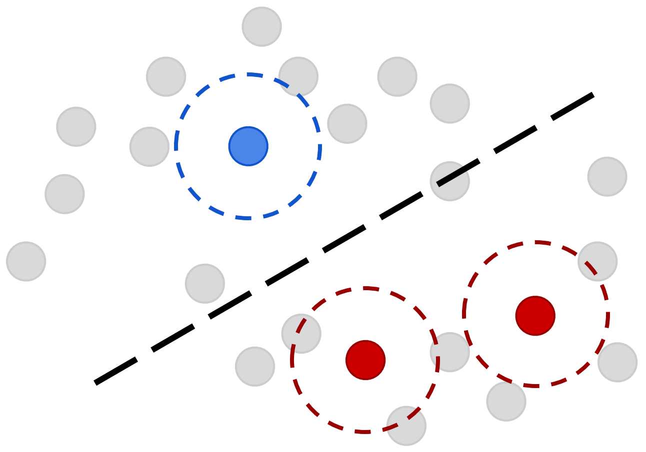
How to better use unlabeled data ?

Vanilla supervised learning



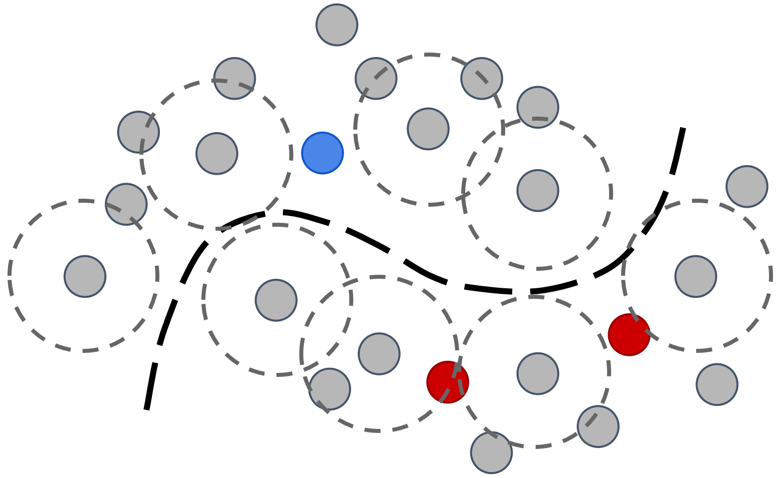
- Considers only *labeled* samples
- Overfits when few training samples

Data augmentation



- Transforms *labeled* samples to augment the training set
- Better generalization, but not enough for semi-supervised learning

Consistency regularization



- Perturbs *unlabeled* samples with noise or guided transformations
- Imposes the network to have consistent outputs for perturbed samples

SSL methods using consistency regularization

Basic transformation consistency (Γ-model)

$$\mathcal{L}(\theta) = \frac{1}{N_l} \sum_{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}) \in \mathcal{D}_l} \ell_{sup}\big(f(\mathbf{X}^{(i)}), \mathbf{Y}^{(i)}\big) + \frac{\lambda}{N_u} \sum_{\mathbf{X}^{(u)} \in \mathcal{D}_u} \mathbb{E}_{T \sim P_T}\Big[\ell_{reg}\big(T(f(\mathbf{X}^{(u)})),\ f(T(\mathbf{X}^{(u)}))\big)\Big]$$

SSL methods using consistency regularization

Basic transformation consistency (Γ-model)

Standard supervised loss

$$\mathcal{L}(\theta) = \boxed{\frac{1}{N_l} \sum_{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}) \in \mathcal{D}_l} \ell_{sup}\left(f(\mathbf{X}^{(i)}), \mathbf{Y}^{(i)}\right)} + \frac{\lambda}{N_u} \sum_{\mathbf{X}^{(u)} \in \mathcal{D}_u} \mathbb{E}_{T \sim P_T}\left[\ell_{reg}\left(T(f(\mathbf{X}^{(u)})),\ f(T(\mathbf{X}^{(u)}))\right)\right]$$

Cross-entropy, Dice, etc.

SSL methods using consistency regularization

Basic transformation consistency (Γ-model)

Transformation consistency loss

$$\mathcal{L}(\theta) = \frac{1}{N_l} \sum_{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}) \in \mathcal{D}_l} \ell_{sup}\big(f(\mathbf{X}^{(i)}), \mathbf{Y}^{(i)}\big) + \boxed{\frac{\lambda}{N_u} \sum_{\mathbf{X}^{(u)} \in \mathcal{D}_u} \mathbb{E}_{T \sim P_T}\Big[\ell_{reg}\big(T(f(\mathbf{X}^{(u)})),\ f(T(\mathbf{X}^{(u)}))\big)\Big]}$$

Random transformation: rotation, flip, crop, etc.

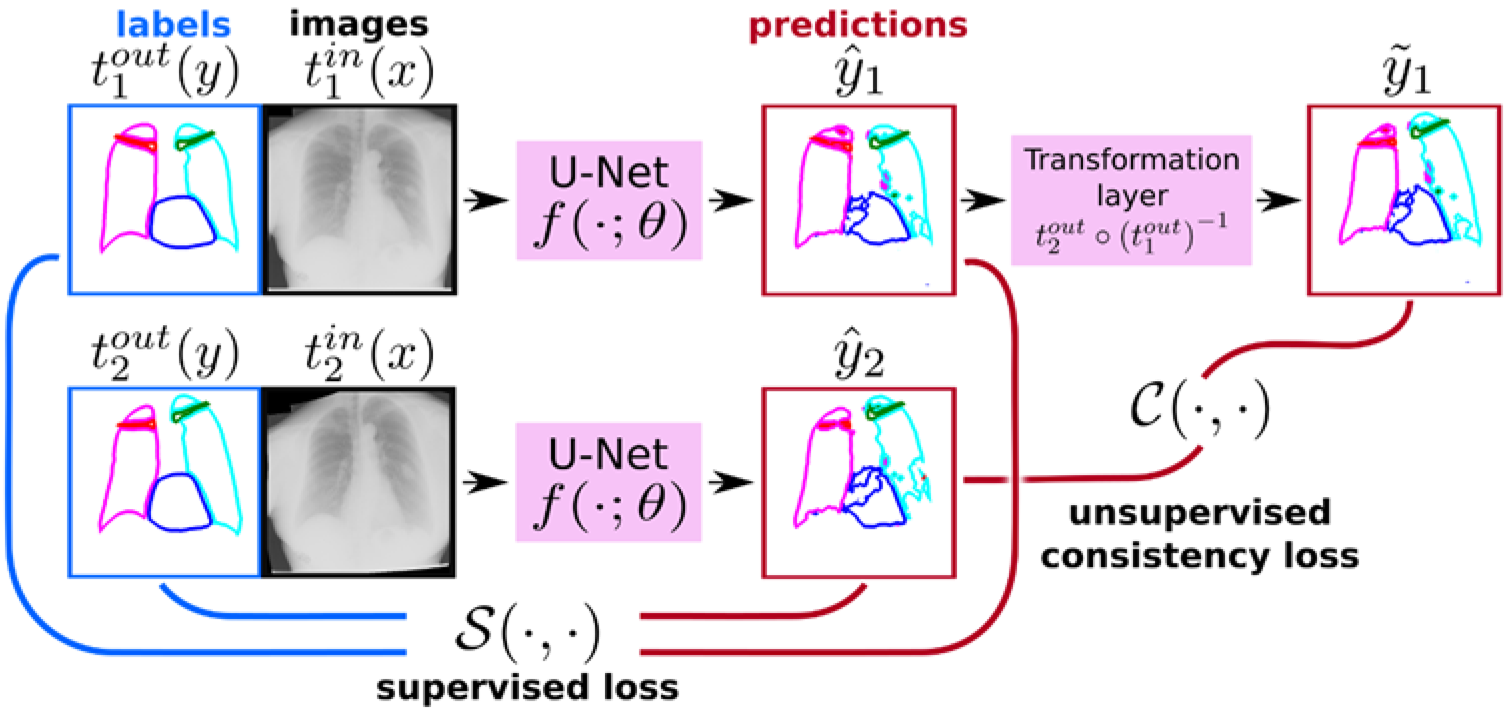
Regularization loss imposing *transformation equivariance*

Regularization losses:

$$\ell_{reg}^{KL}(\mathbf{S}, \mathbf{S}') = \sum_{p \in \Omega} \sum_{k \in \mathcal{C}} s_{p,k} \log\left(\frac{s_{p,k}}{s'_{p,k}}\right) \qquad \ell_{reg}^{L_2}(\mathbf{S}, \mathbf{S}') = \sum_{p \in \Omega} \|\mathbf{s}_p - \mathbf{s}'_p\|_2^2$$

SSL methods using consistency regularization

Application to chest X-ray segmentation:

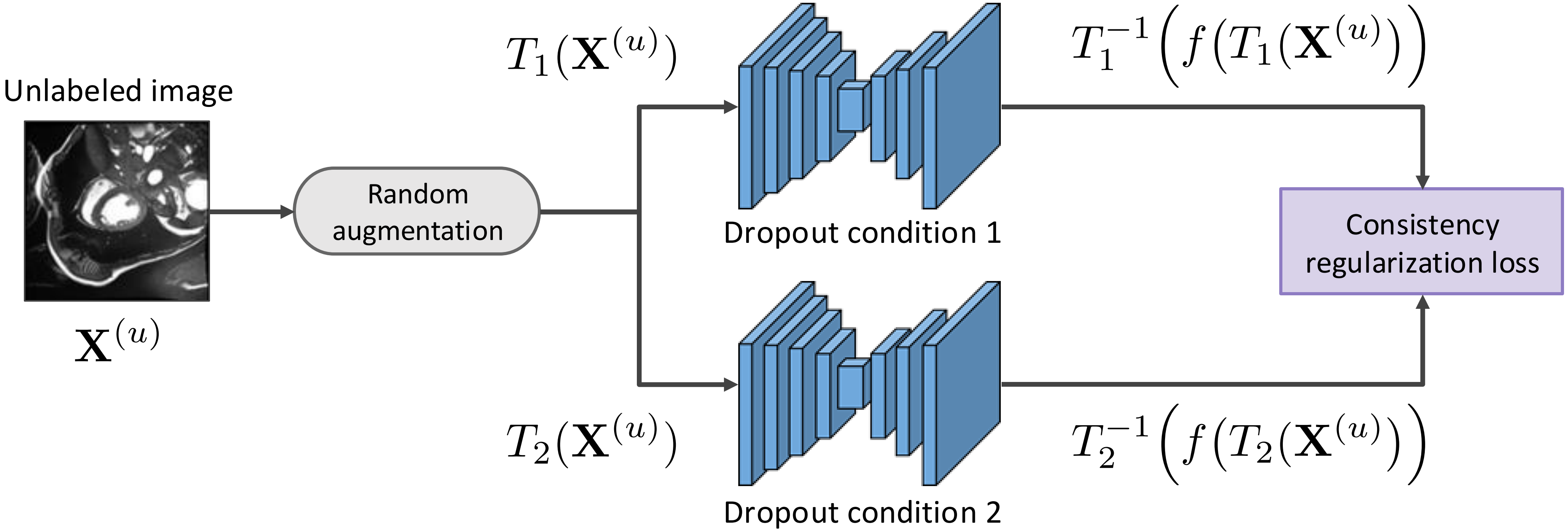


Transformations are random elastic deformations

Bortsova, G., et al.. "Semi-supervised medical image segmentation via learning consistency under transformations." MICCAI, 2019.

SSL methods using consistency regularization

Self-ensembling (Π-model):

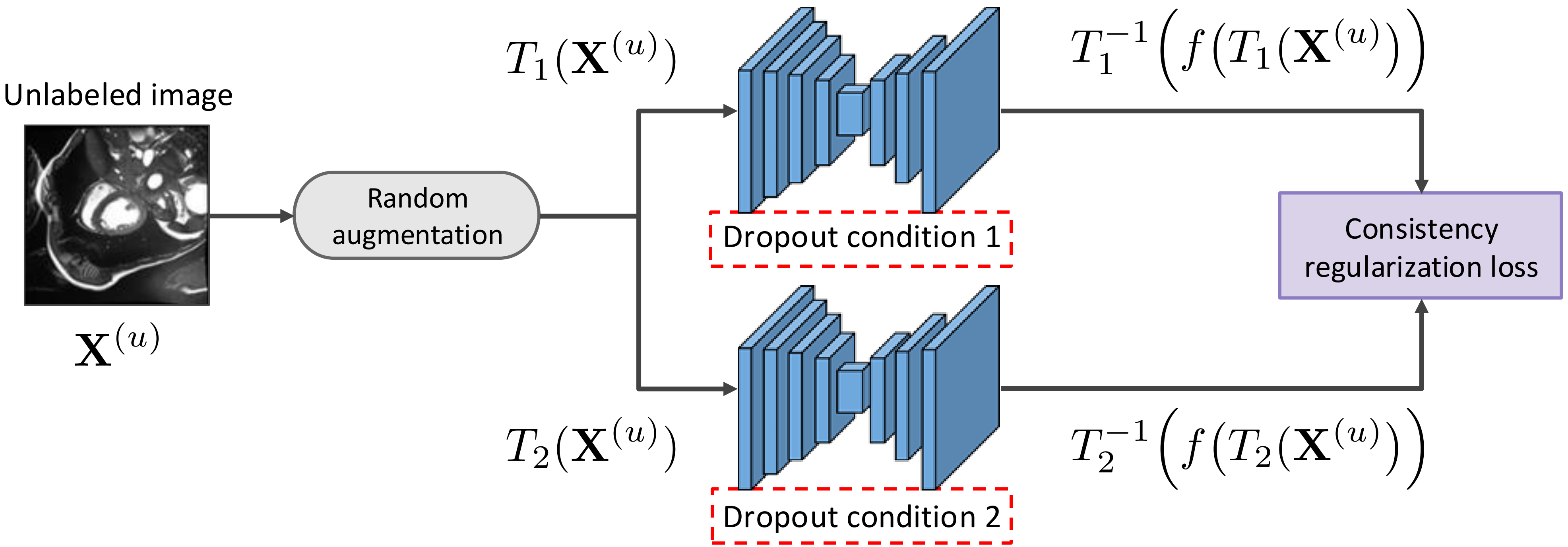


Key idea:

- Applying <u>different dropouts</u> on the <u>same network</u> gives an ensemble of models
- Also leverages random image transformations

SSL methods using consistency regularization

Self-ensembling (Π-model):

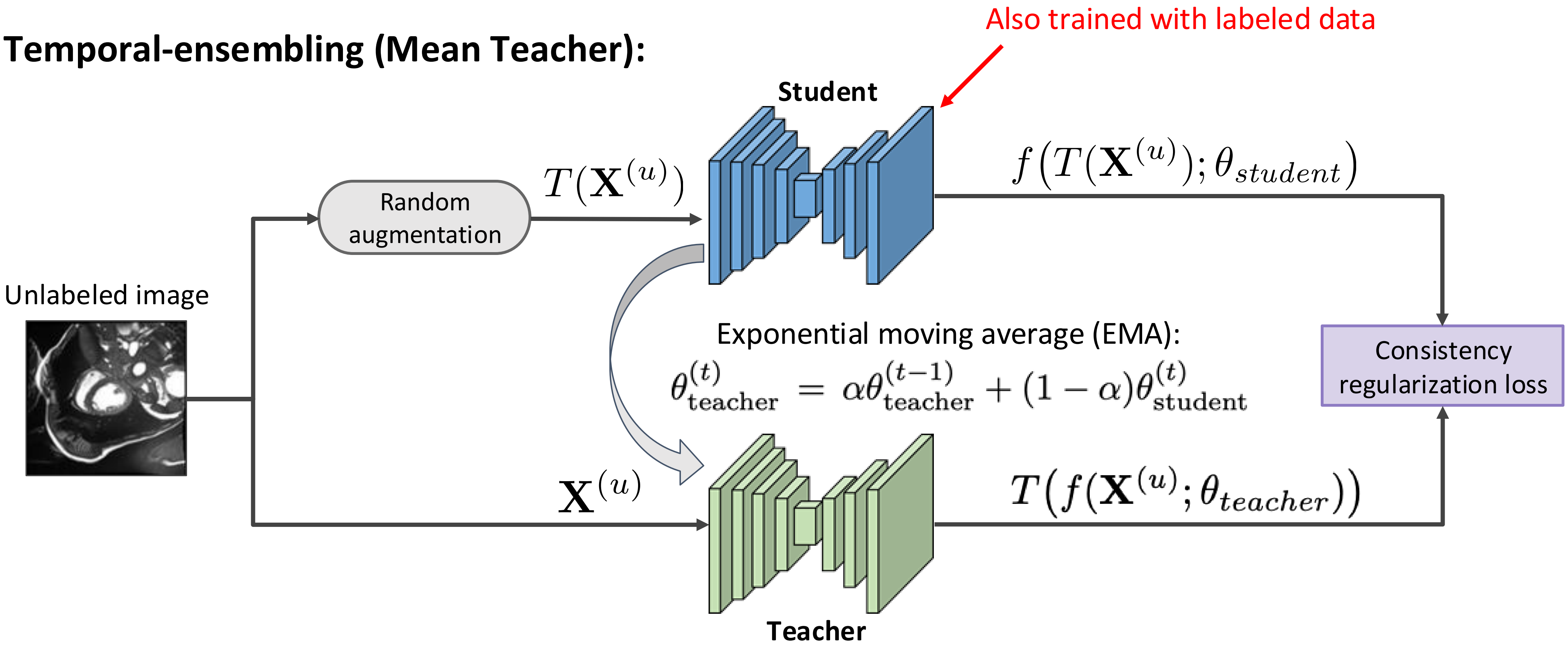


Key idea:

- Applying <u>different dropouts</u> on the <u>same network</u> gives an ensemble of models
- Also leverages random image transformations

SSL methods using consistency regularization

Temporal-ensembling (Mean Teacher):

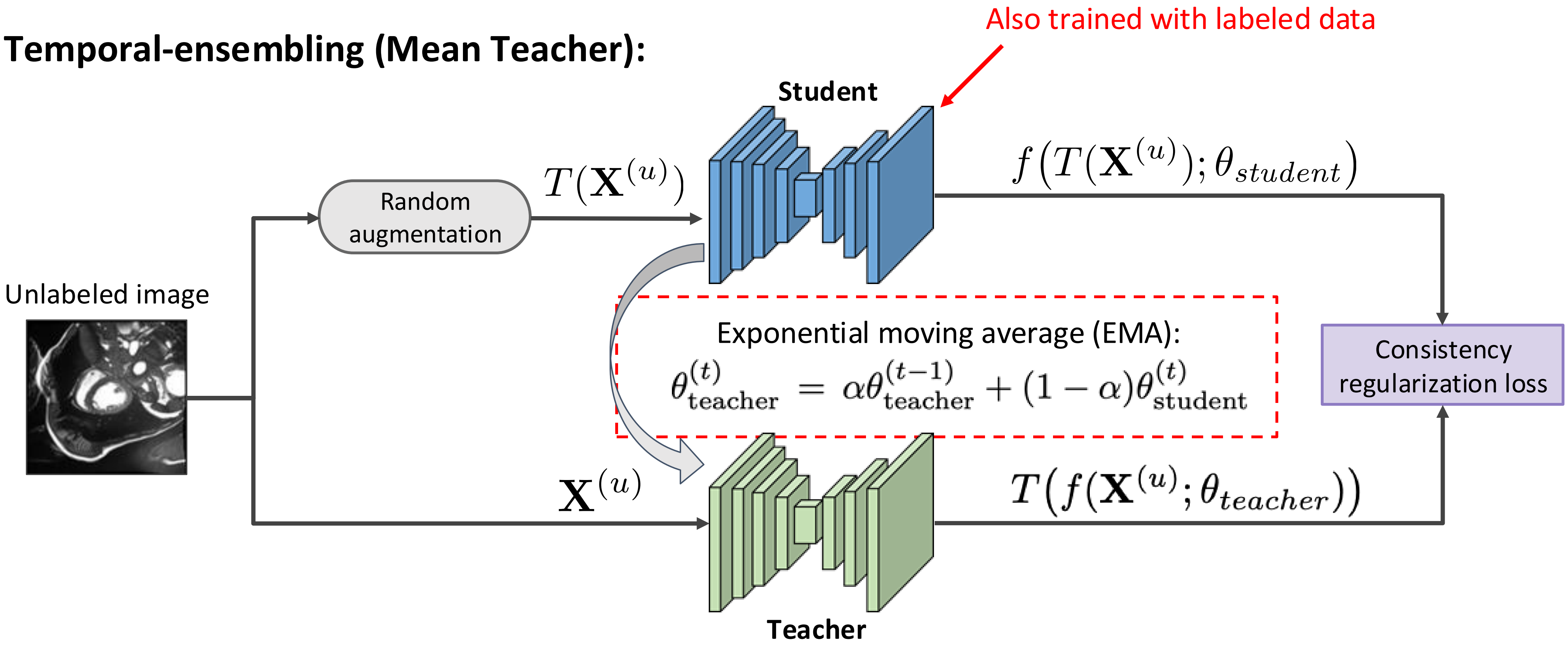


Key idea:

- Consistency between the predictions of a Teacher and a Student network
- The Teacher's weights are an EMA of the Student's at previous training iterations ($\alpha \approx 1$)

SSL methods using consistency regularization

Temporal-ensembling (Mean Teacher):

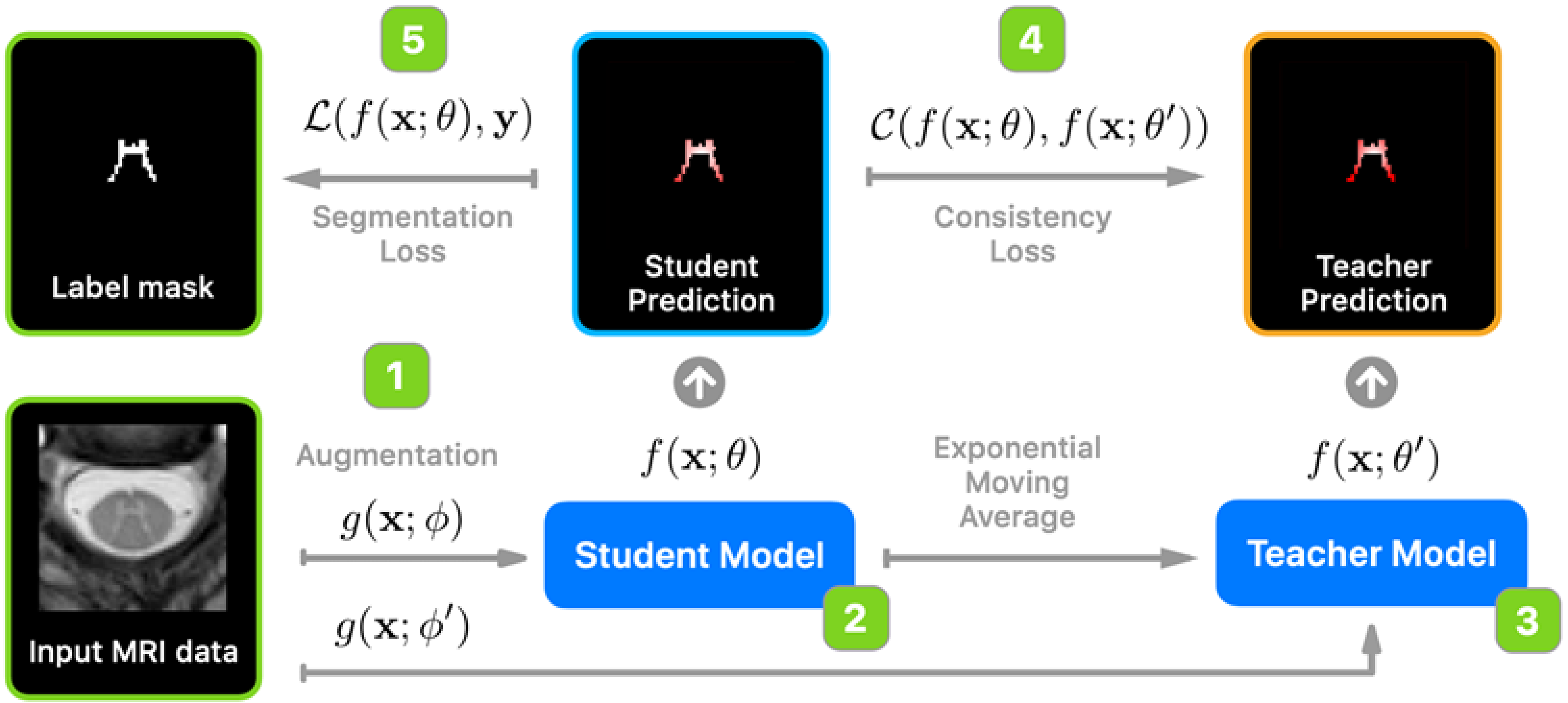


Key idea:

- Consistency between the predictions of a Teacher and a Student network
- The Teacher's weights are an EMA of the Student's at previous training iterations ($\alpha \approx 1$)

SSL methods using consistency regularization

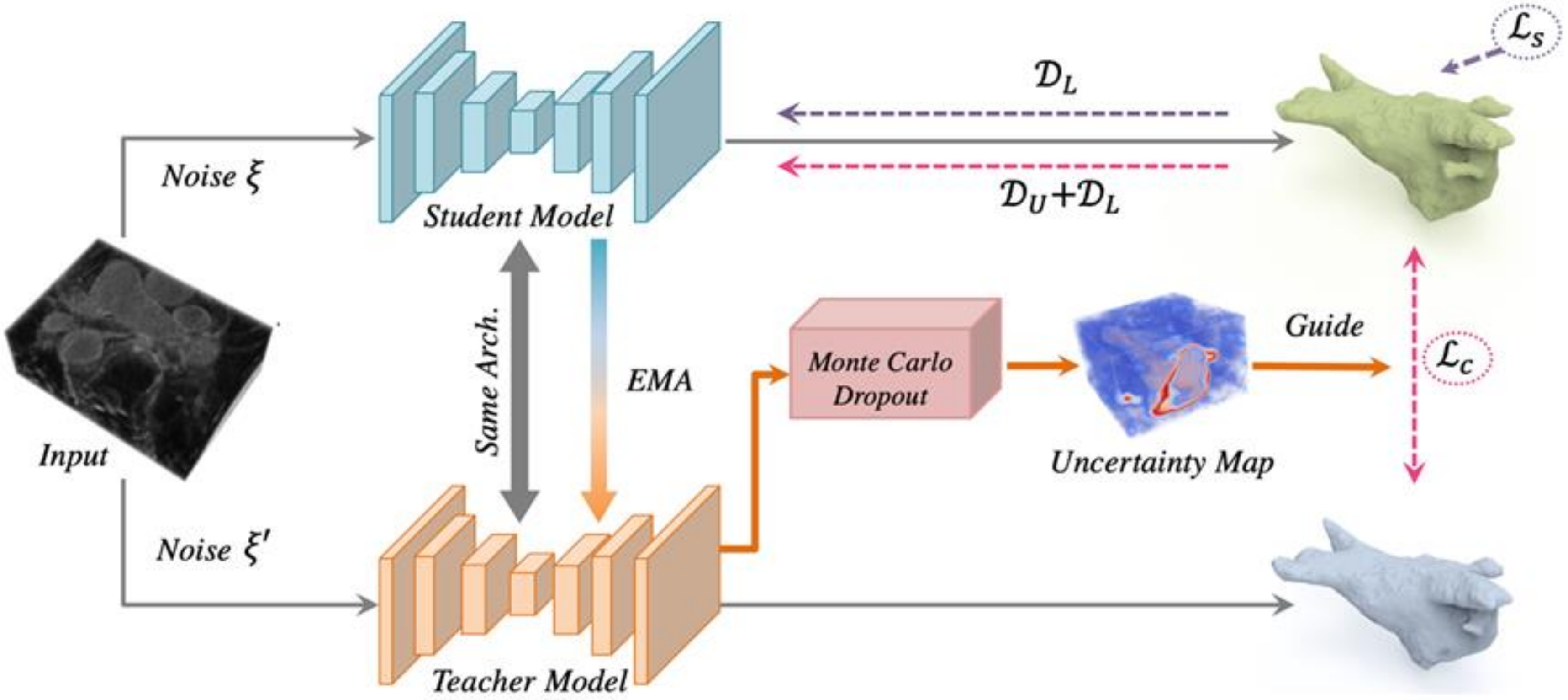
Application of Mean Teacher to segmenting MRI spinal cord gray matter



Perone, C.S. and Cohen-Adad, J. "Deep semi-supervised segmentation with weight-averaged consistency targets." Deep learning in medical image analysis and multimodal learning for clinical decision support, 2019 (*extended version in Neuroimage*)

SSL methods using consistency regularization

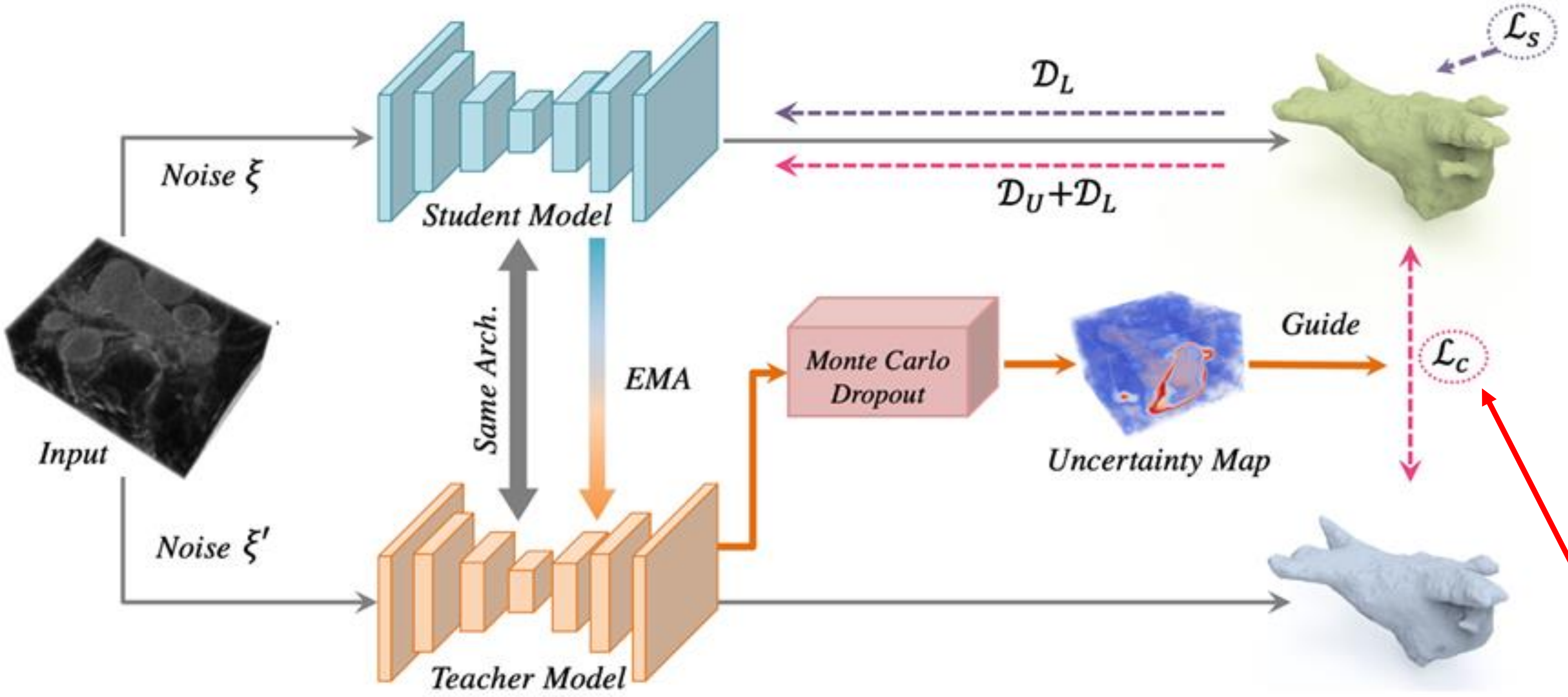
Uncertainty-aware self-ensembling



Yu, L et al.. "Uncertainty-aware self-ensembling model for semi-supervised 3D left atrium segmentation." MICCAI, 2019.

SSL methods using consistency regularization

Uncertainty-aware self-ensembling



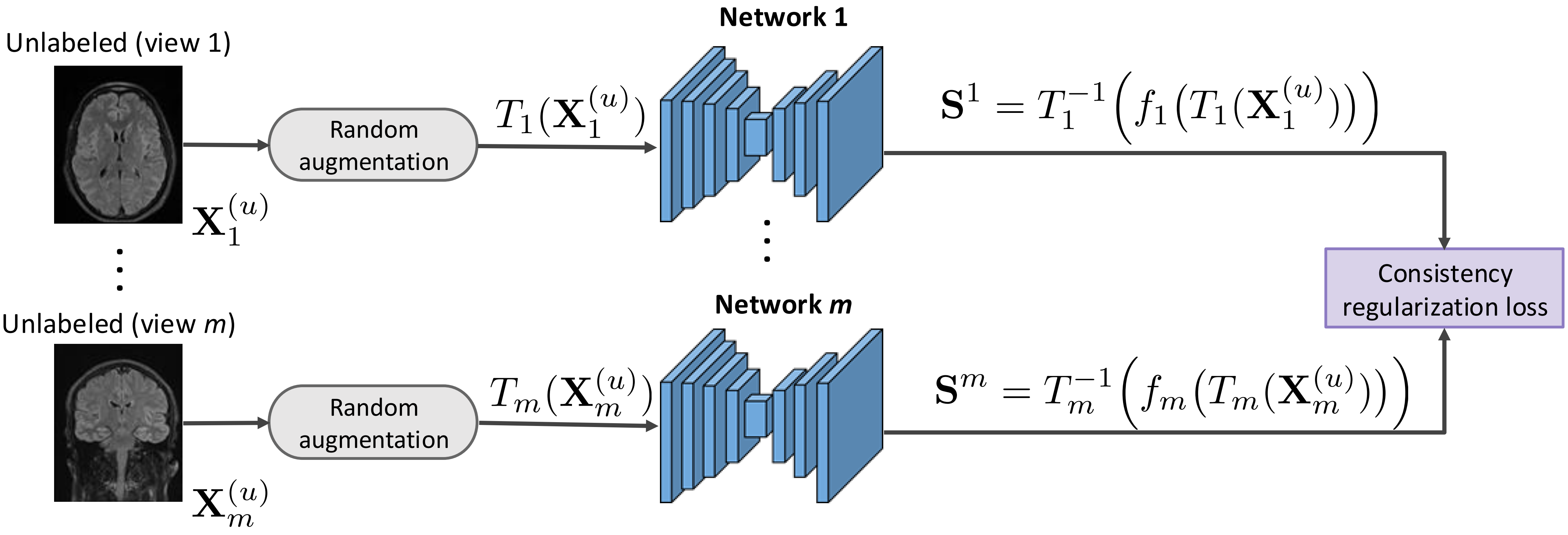
Enforces consistency only in low-uncertainty regions of the image

$$\mathcal{L}_c(f', f) = \frac{\sum_v \mathbb{I}(u_v < H) \| f'_v - f_v \|^2}{\sum_v \mathbb{I}(u_v < H)},$$

Yu, L et al.. "Uncertainty-aware self-ensembling model for semi-supervised 3D left atrium segmentation." MICCAI, 2019.

SSL methods using consistency regularization

Muti-view co-training

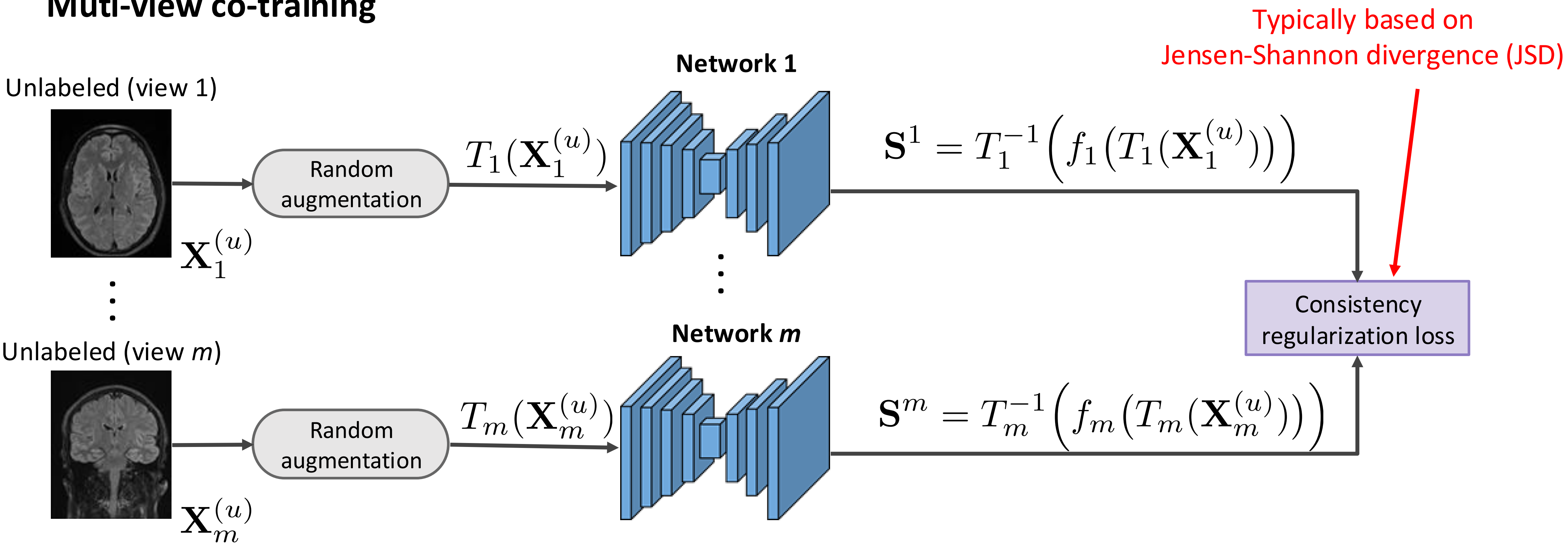


Key idea:

- Supposes the existence of separate, complementary views of the data
- Use high-confidence predictions for a given view as pseudo-labels in other views

SSL methods using consistency regularization

Muti-view co-training

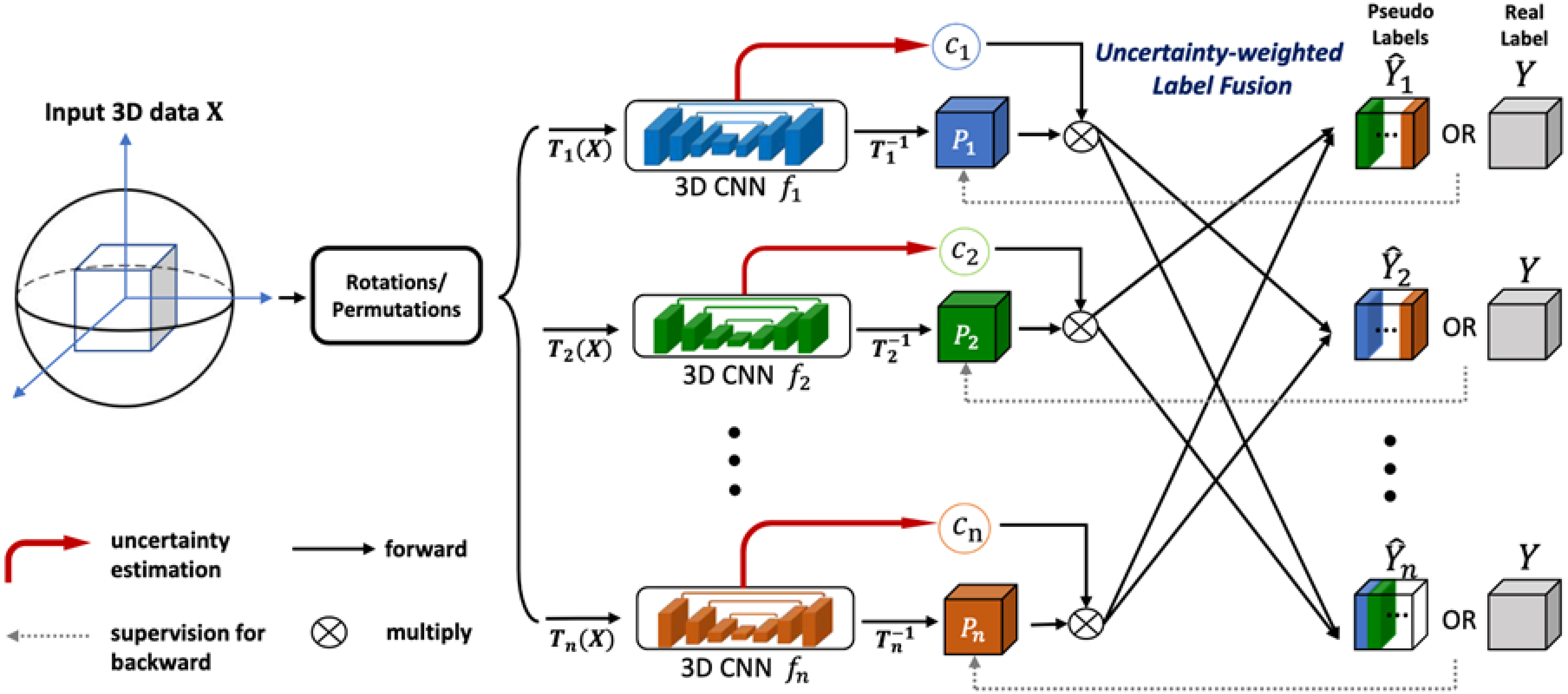


Key idea:

- Supposes the existence of separate, complementary views of the data
- Use high-confidence predictions for a given view as pseudo-labels in other views

SSL methods using consistency regularization

Application of multi-view co-training for pancreas and liver tumor segmentation



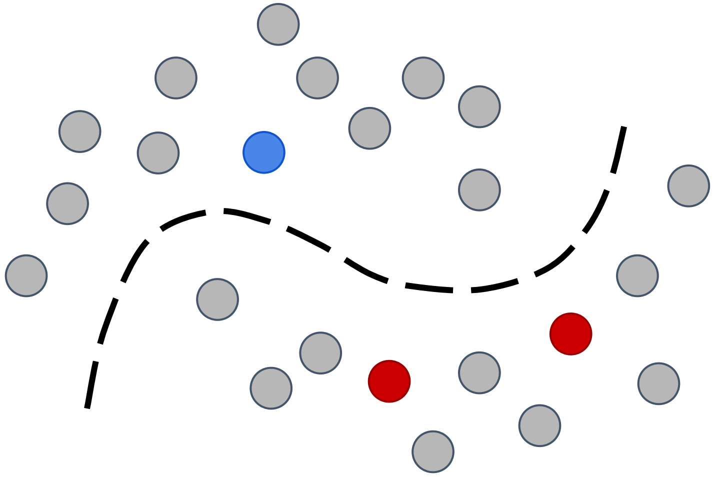
Xia, Y., et al. "3D semi-supervised learning with uncertainty-aware multi-view co-training." WACV 2020.

Self-supervised learning

for segmentation

Self-supervised representation learning

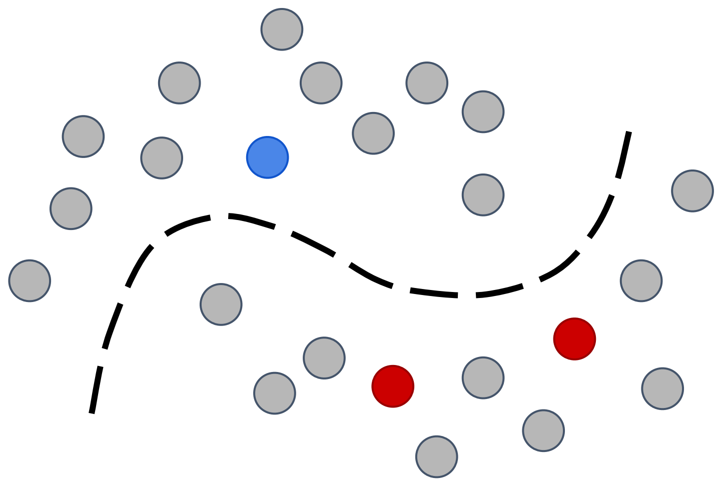
Traditional semi-supervised learning



- Trains a model simultaneously with both labeled and unlabeled data

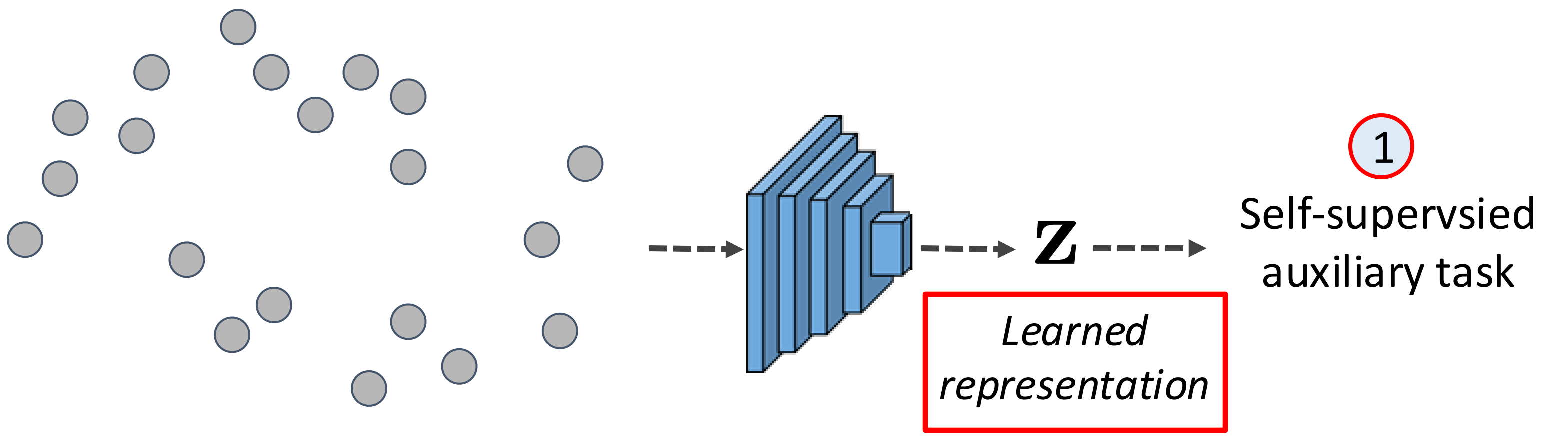
Self-supervised representation learning

Traditional semi-supervised learning



- Trains a model simultaneously with both labeled and unlabeled data

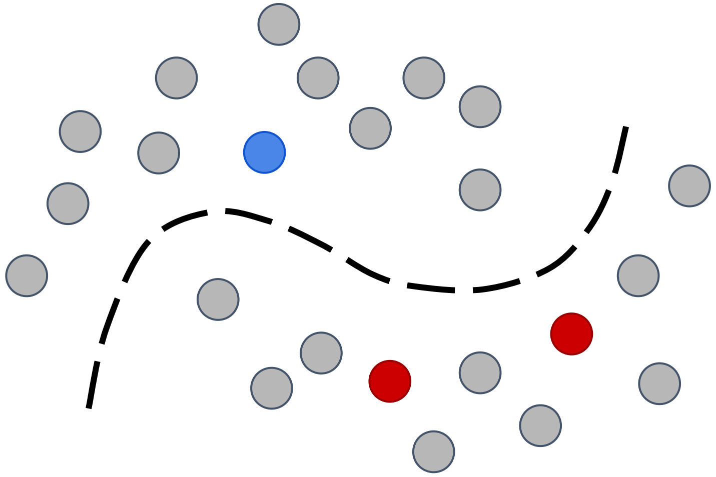
Self-supervised representation learning



- Learns a data representation by solving an auxiliary task that <u>does not require labels</u>
- Uses this representation to solve a downstream task
- A light weight fine-tuning of the model can generally be done

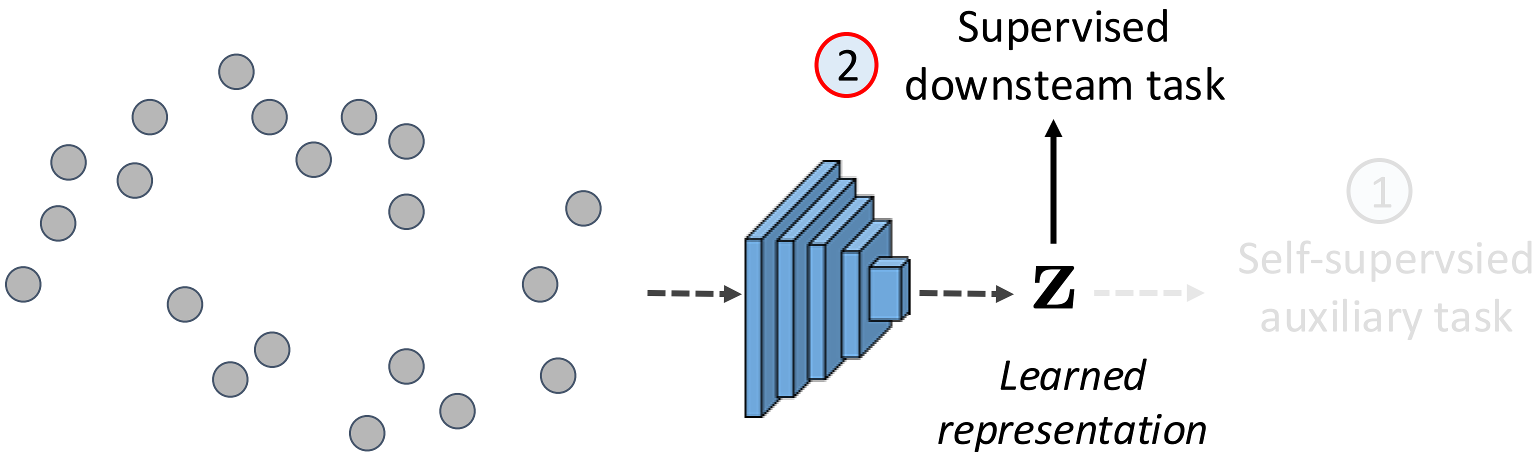
Self-supervised representation learning

Traditional semi-supervised learning



- Trains a model simultaneously with both labeled and unlabeled data

Self-supervised representation learning

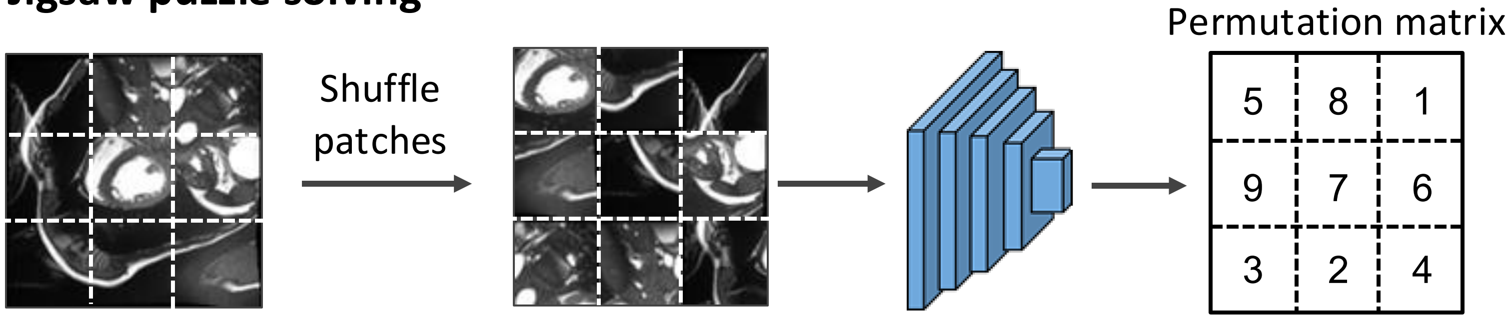


- Learns a data representation by solving an auxiliary task that <u>does not require labels</u>
- Uses this representation to solve a downstream task
- A light weight fine-tuning of the model can generally be done

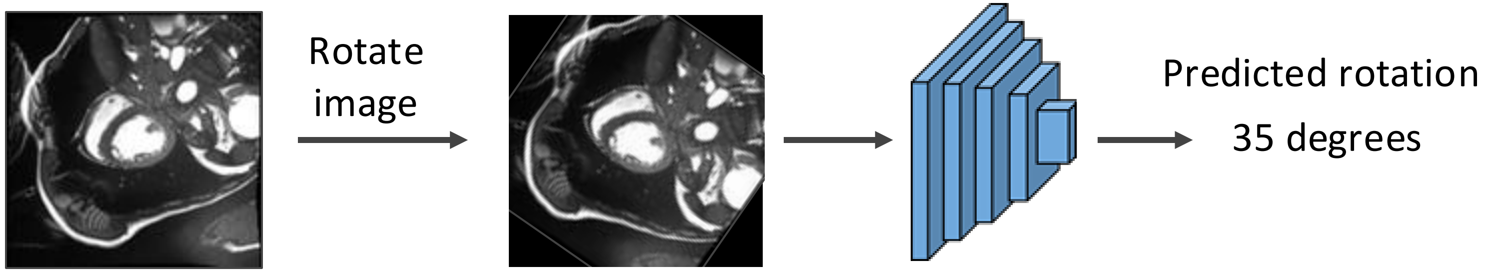
Approaches for URL

Self-supervised learning:

Jigsaw puzzle solving



Rotation prediction



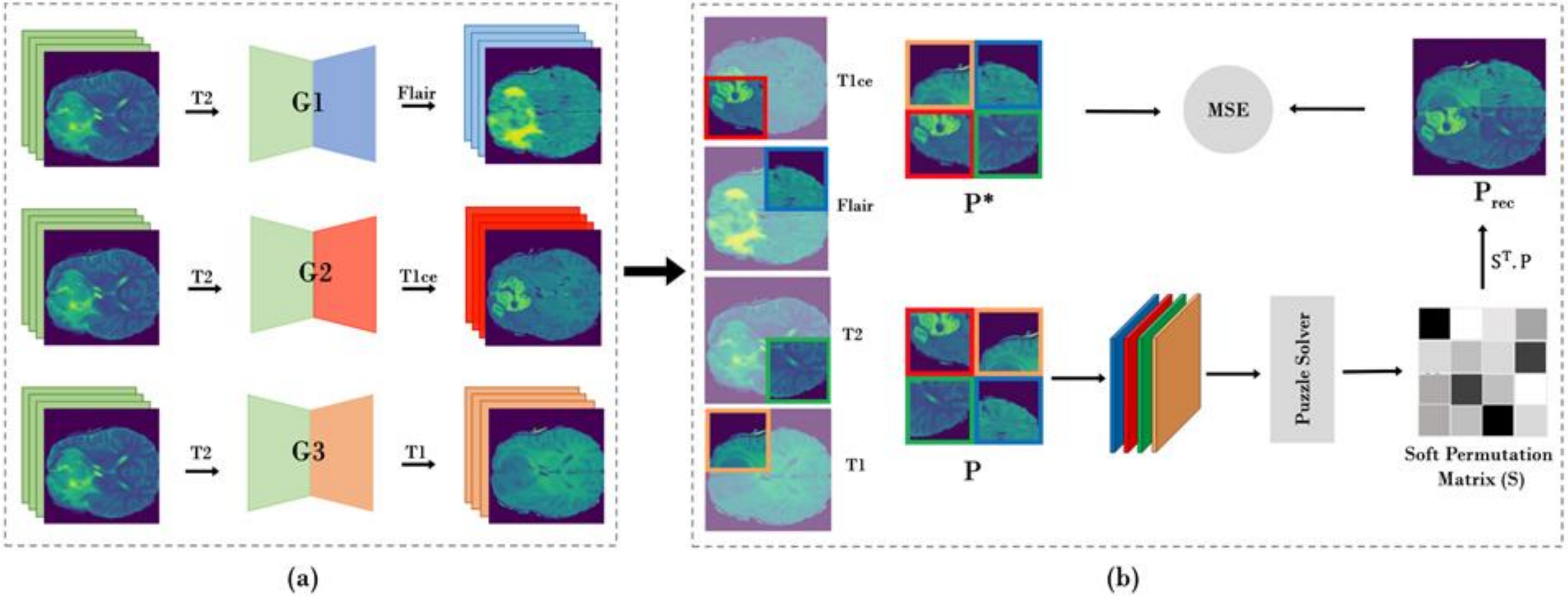
Basic idea:

- Learn to solve a pretext task which does not require annotations
- <u>Example</u>: find the correct order of permuted patches (*see left*)

Taleb, A., et al. "Multimodal self-supervised learning for medical image analysis." arXiv preprint (2019).

Approaches for URL

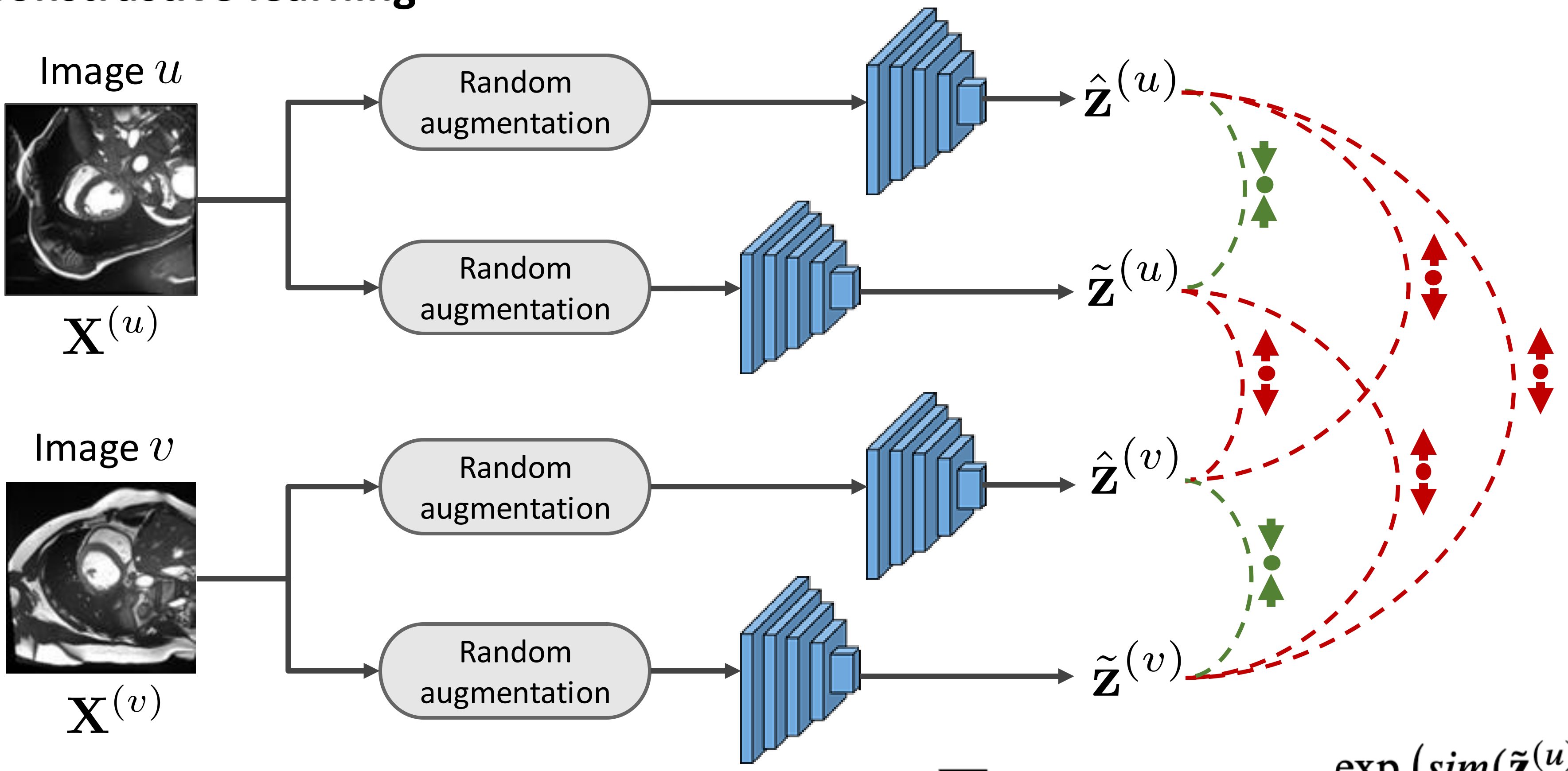
Application to brain tumor MRI



Taleb, A., et al. "Multimodal self-supervised learning for medical image analysis." arXiv preprint (2019).

Self-supervised representation learning

Constrastive learning



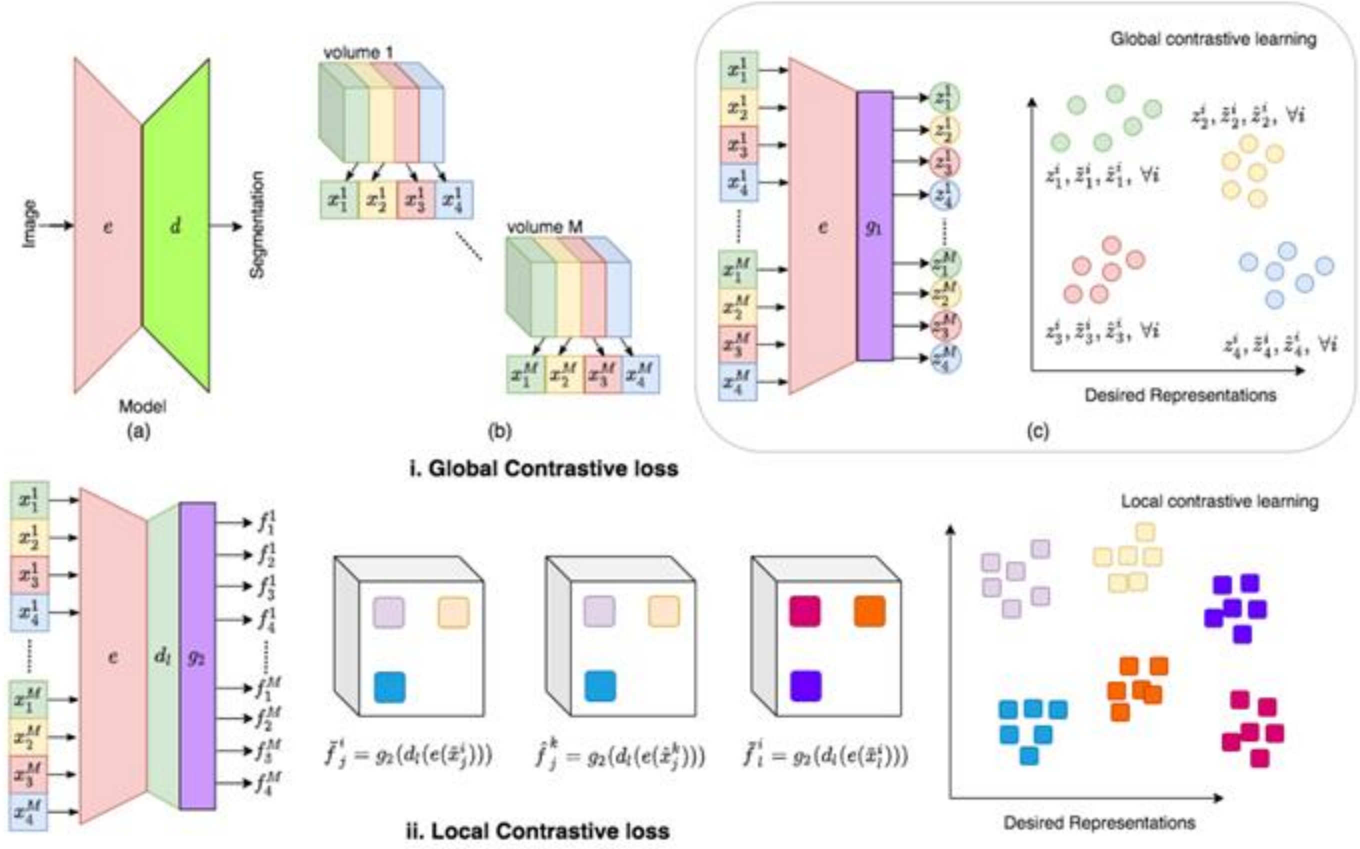
$$\mathcal{L}_{SimCLR}(\theta) = -\sum_{u} \log \frac{\exp\left(sim(\tilde{\mathbf{z}}^{(u)}, \hat{\mathbf{z}}^{(u)})/\tau\right)}{\sum_{v \neq u} \exp\left(sim(\tilde{\mathbf{z}}^{(u)}, \tilde{\mathbf{z}}^{(v)})/\tau\right) + \exp\left(sim(\tilde{\mathbf{z}}^{(u)}, \hat{\mathbf{z}}^{(v)})/\tau\right)}$$

Key idea:

- Generate two augmentations for each image
- Pull closer the representations of the same image and push away those of different ones

Approaches for URL

Contrastive learning:



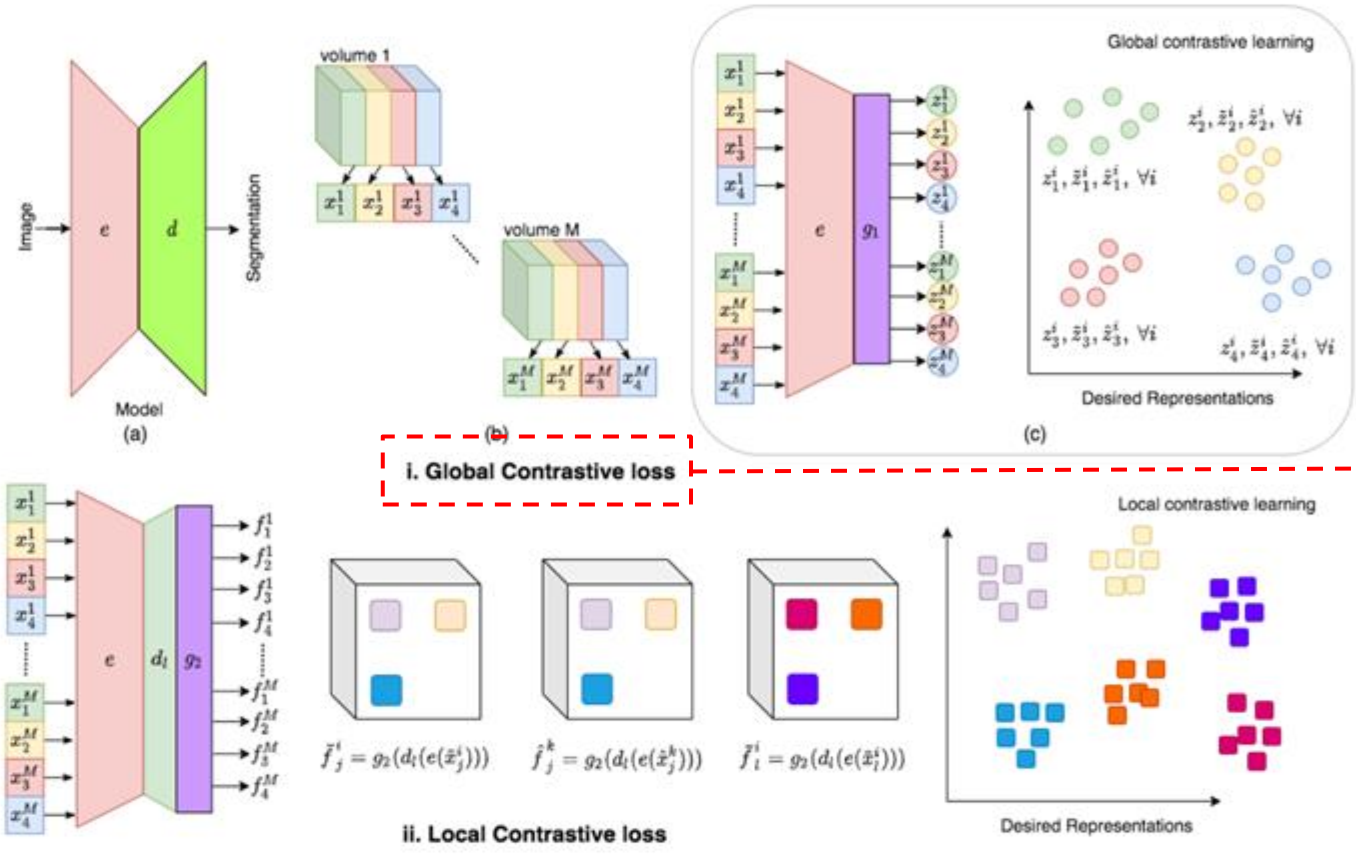
Basic idea:

- Train with pairs of images that match (e.g., same position in volume, same image under different transformations, etc.) or not
- Find a representation that is similar for matching pairs and different for non-matching ones

Chaitanya, K., et al. "Contrastive learning of global and local features for medical image segmentation with limited annotations." NeurIPS 2020.

Approaches for URL

Contrastive learning:



Global contrastive loss

$$-\log \frac{\boxed{e^{\mathrm{sim}(\tilde{z},\hat{z})/\tau}}}{\boxed{e^{\mathrm{sim}(\tilde{z},\hat{z})/\tau}} + \boxed{\sum_{\bar{x}\in\Lambda^-} e^{\mathrm{sim}(\tilde{z},g_1(e(\bar{x})))/\tau}}}$$

$$\tilde{z} = g_1(e(\tilde{x})),\ \hat{z} = g_1(e(\hat{x})).$$

Matching pair

Non-matching pairs

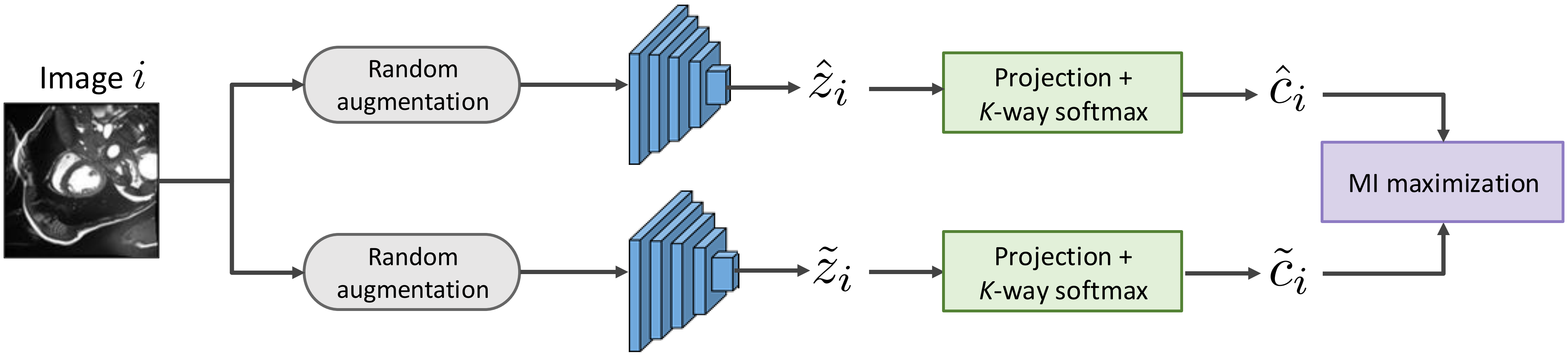
Basic idea:

- Matching pairs for global loss are slices in the same volume position or same subject
- Matching pairs for local loss are feature vectors from the same image under two transformations

Chaitanya, K., et al. "Contrastive learning of global and local features for medical image segmentation with limited annotations." NeurIPS 2020.

Self-supervised representation learning

Clustering based on mutual information (MI)



$$\mathrm{MI} = D_{\mathrm{KL}}\big(P(\hat{c}_i, \tilde{c}_i) \,\|\, P(\hat{c}_i) \cdot P(\tilde{c}_i)\big)$$

Key idea:

- Project features to a discrete K-cluster probability distribution
- Enforce transformation invariance to clusters using MI maximization

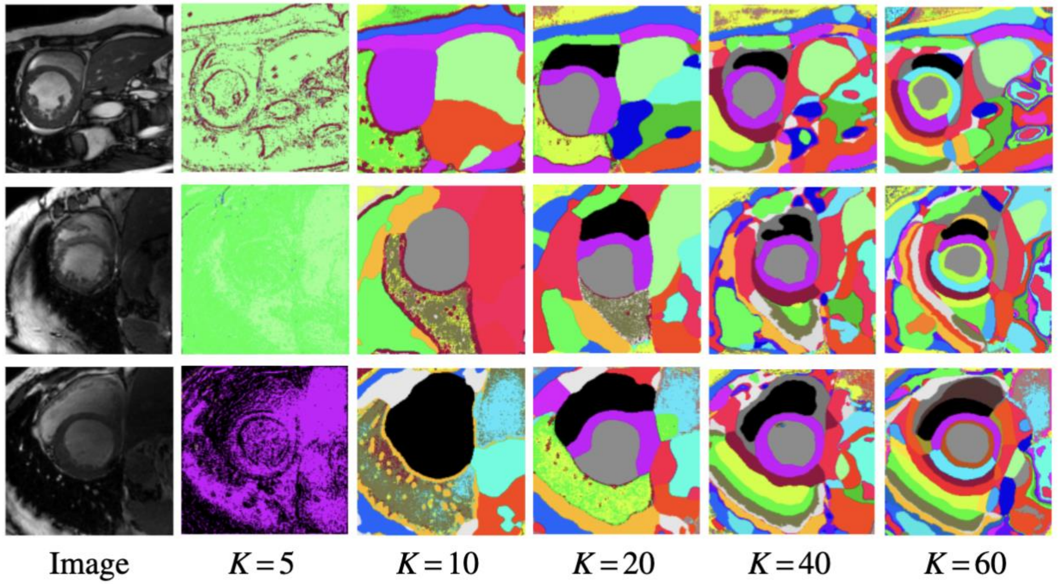
Boundary-aware Information Maximization for Self-supervised Medical Image Segmentation

Jizong Peng*
ETS Montreal
jizong.peng.1@etsmtl.net

Ping Wang
ETS Montreal
ping.wang.1@ens.etsmtl.ca

Christian Desrosiers
ETS Montreal
christian.desrosiers@etsmtl.ca

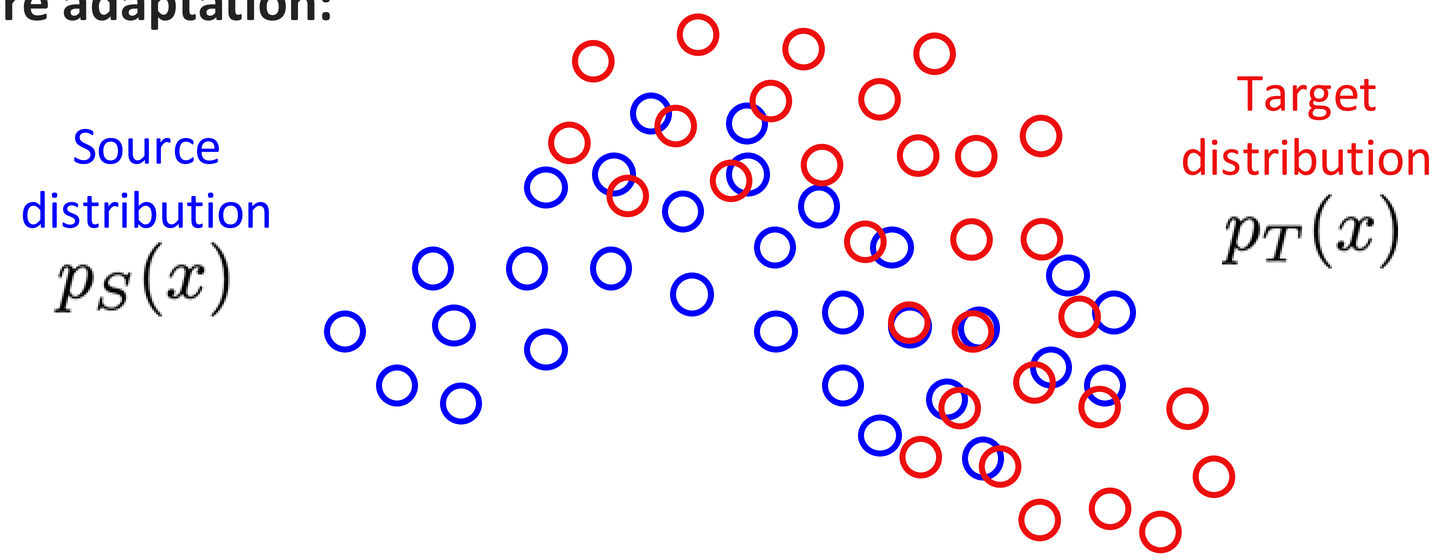
Marco Pedersoli
ETS Montreal
marco.pedersoli@etsmtl.ca



Methods	ACDC-LV				ACDC-RV				ACDC-Myo				PROMISE12			
	n=1	*n*=2	*n*=4	avg	*n*=1	*n*=2	*n*=4	avg	*n*=1	*n*=2	*n*=4	avg	*n*=4	*n*=6	*n*=8	avg
Partial supervision	67.13	74.49	84.81	75.48	51.82	60.50	64.18	58.84	54.05	67.56	76.00	65.87	49.91	71.53	78.04	66.49
Full supervision	92.26				86.80				88.07				89.65			
Contrast (*Enc+Dec*)	77.98	85.97	88.42	84.12	66.47	72.82	76.69	71.99	64.96	76.98	78.76	73.57	60.68	77.97	80.53	73.06
Ours (*pre-train*)	84.48	87.85	90.04	87.45	75.42	79.73	78.89	78.01	74.30	78.43	82.82	78.52	69.76	80.47	82.09	77.44
Entropy minimization	73.79	80.26	86.84	80.30	56.18	62.09	66.27	61.51	57.23	71.10	76.28	68.20	59.78	76.09	78.98	71.62
MixUp	73.30	76.30	84.42	78.01	61.23	63.60	63.14	62.66	55.74	69.80	73.84	66.46	52.09	75.59	81.11	69.60
Mean Teacher (MT)	83.13	87.02	87.70	85.95	61.61	68.76	67.21	65.86	61.55	75.32	78.42	71.76	84.71	85.97	86.93	85.87
UA-MT	81.08	85.03	87.19	84.43	62.06	67.91	66.64	65.54	59.26	73.68	78.61	70.52	66.16	81.79	84.40	77.45
ICT	76.87	78.41	86.34	80.54	60.31	63.42	68.35	64.03	55.91	71.77	77.90	68.53	63.97	77.92	81.39	74.43
Adversarial learning	75.31	74.85	85.85	78.67	55.29	62.25	64.58	60.71	57.68	70.39	75.94	68.00	71.50	78.63	81.35	77.16
MT + Contrast (*Enc+Dec*)	86.37	89.57	90.40	88.78	75.53	78.42	77.22	77.06	76.11	80.21	82.00	79.44	76.16	82.89	84.85	81.30
MT + Ours (*pre-train*)	**90.25**	**91.36**	**91.04**	**90.88**	**80.16**	**81.50**	**78.97**	**80.21**	**78.71**	**83.33**	**83.61**	**81.88**	**85.64**	**85.60**	**88.45**	**86.56**

Domain adaptation

Before adaptation:



Domain adaptation

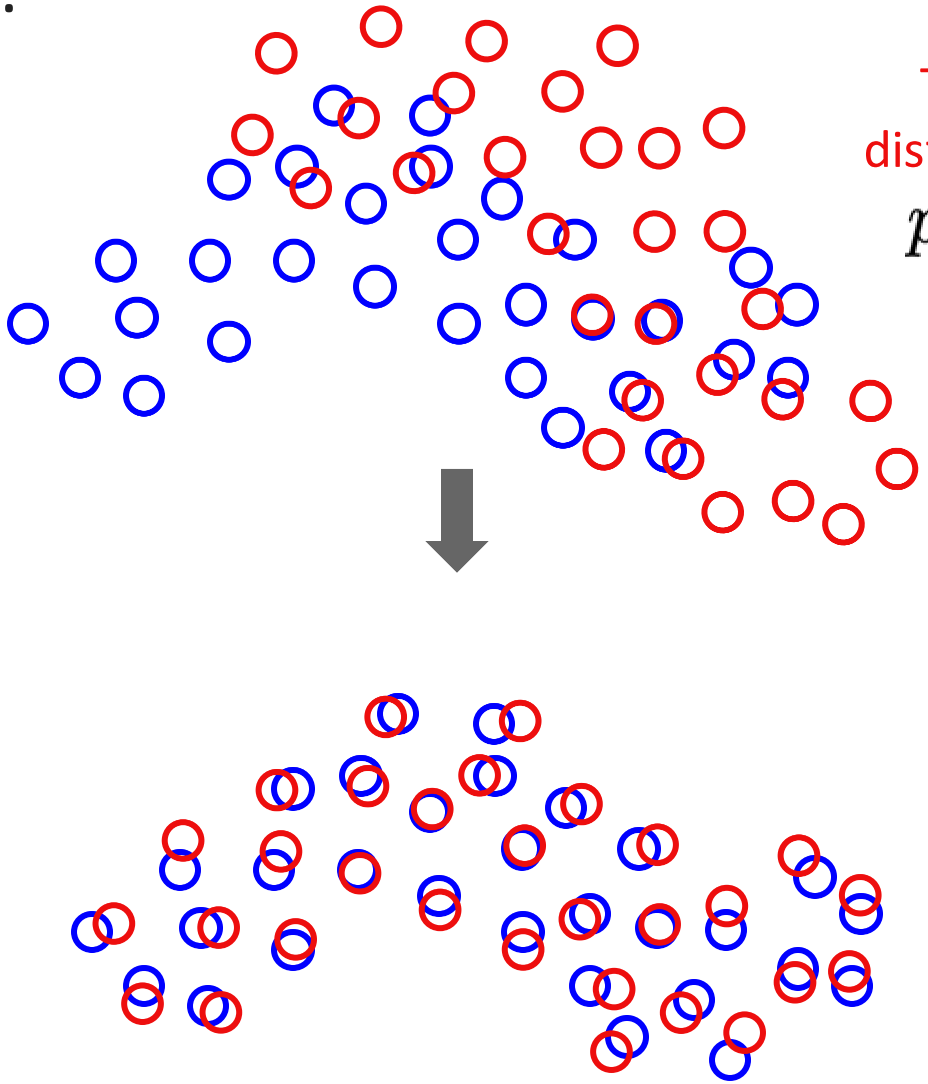
Before adaptation:

Source distribution

$p_S(x)$

Target distribution

$p_T(x)$



After adaptation:

$$p_S(x) = p_T(x)$$

Objective

Align the distributions (input, output or representation) so that a model trained on Source data also works on Target data

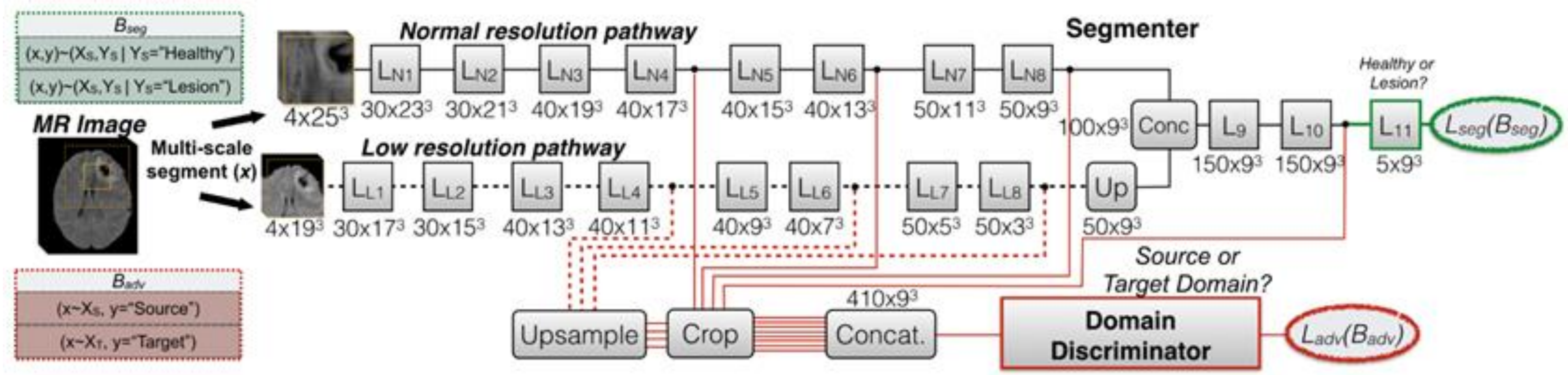
Unsupervised domain adaptation (UDA)

Unsupervised domain adaptation (UDA)

- Like semi-supervised segmentation except target images are **<u>from a different domain</u>**
- Most semi-supervised learning methods (*adversial learning, pseudo-label, consistency learning, etc.*) **<u>can also be used for UDA</u>**

Adversarial domain adaptation

Adversarial domain adaptation for brain lesion segmentation



Source domain (Database 1):

- GE, FLAIR, T2, MPRAGE, PD

Target domain (Database 2):

- SWI, FLAIR, T2, MPRAGE, PD

No adaptation (All sequences) | No adaptation (No GE/SWI) | Adaptation S→T | Ground-truth

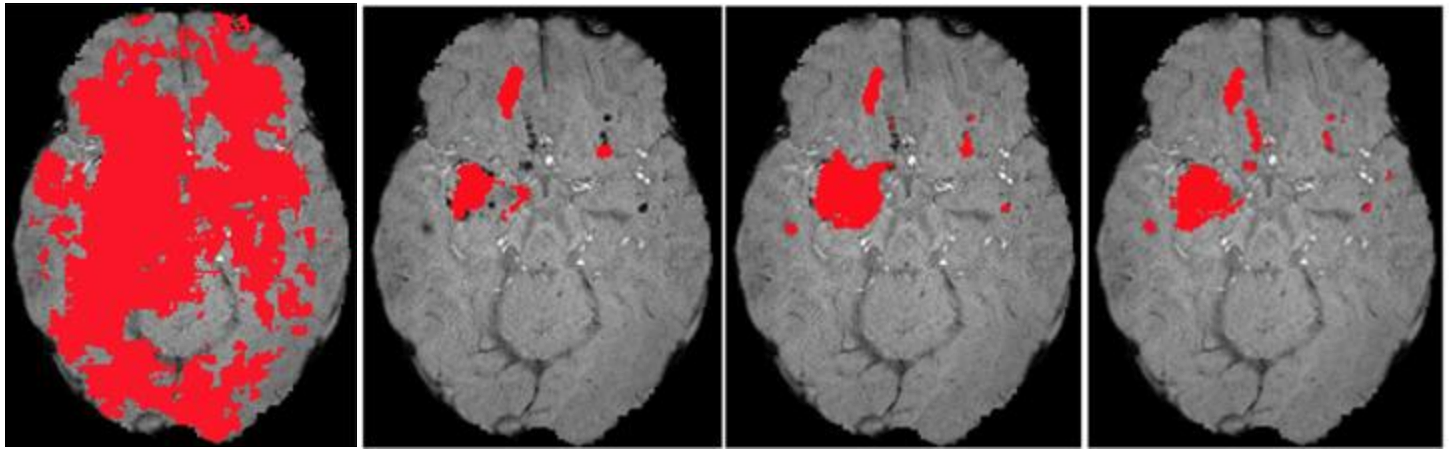


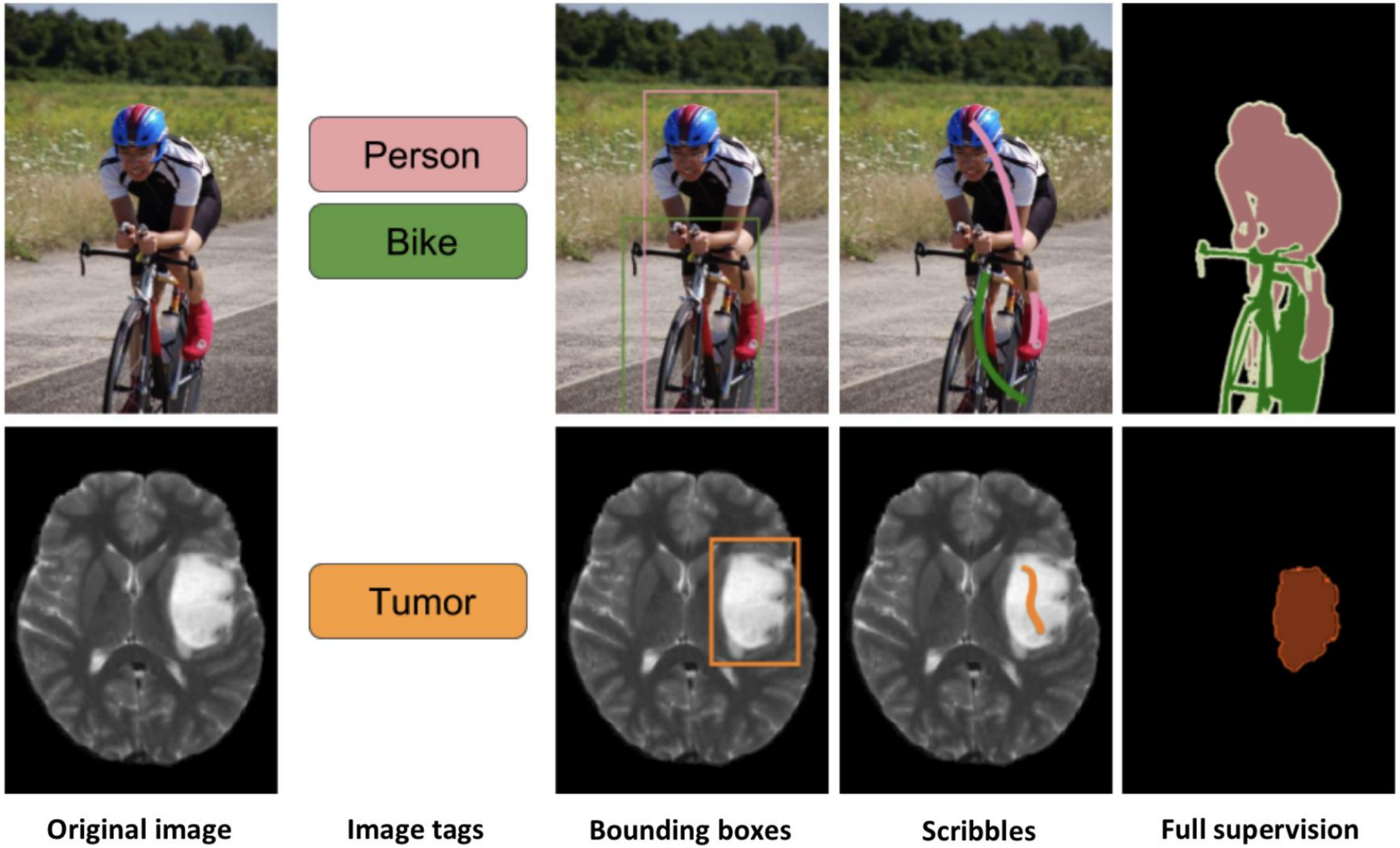
Image from Kamnitsas, K., et al. "Unsupervised domain adaptation in brain lesion segmentation with adversarial networks." *Int. Conf. on Information Processing in Medical Imaging*, 2017.

Part II:

Weakly-supervised learning

From full to weak supervision

Different levels of supervision



Idea:

Weak labels like image tags, bounding boxes and scribbles are cheaper to obtain

Challenge:

How can we use this partial information to train the segmentation network?

Image from Dolz et al., Less-supervised Segmentation with CNNs, MICCAI Book Series, Springer 2025

Pixel-level annotations

Weakly-supervised loss:

$$\min_{\theta} \sum_{i=1}^{N} \sum_{p \in \Omega^{(i)}} \ell_s(\mathbf{y}_p^{(i)}, \mathbf{s}_p^{(i)}) + \lambda \sum_{i=1}^{N} \ell_u(\mathbf{S}^{(i)})$$

Key idea:

- For each image i, class labels are only known for a reduced set of pixels $\mathbf{\Omega}^{(i)} \subset \mathbf{\Omega}$
- Include a segmention prior in the form of a regularization loss

Pixel-level annotations

Weakly-supervised loss:

$$\min_{\theta} \sum_{i=1}^{N} \underbrace{\sum_{p \in \Omega^{(i)}}}_{\text{Labeled pixels of image } i} \overbrace{\ell_s}^{\text{Standard loss (e.g., cross-entropy)}}(\mathbf{y}_p^{(i)}, \mathbf{s}_p^{(i)}) + \lambda \sum_{i=1}^{N} \underbrace{\boxed{\ell_u(\mathbf{S}^{(i)})}}_{\text{Regularization loss (segmentation prior)}}$$

Key idea:

- For each image i, class labels are only known for a reduced set of pixels $\Omega^{(i)} \subset \Omega$
- Include a segmention prior in the form of a regularization loss

Examples of regularization losses

Entropy minimization:

$$\ell_u(\mathbf{S}^{(i)}) = -\sum_{p \in \Omega \setminus \Omega^{(i)}} \sum_{k \in \mathcal{C}} s_{p,k}^{(i)} \log s_{p,k}^{(i)}$$

Key idea:

- Push the predictions for unlabled **pixels** to be confident

Examples of regularization losses

Conditional random field (CRF):

$$\ell_u(\mathbf{S}^{(i)}) = \sum_{p,q \in \Omega} w_{pq} \; \ell_{\text{CRF}}(\mathbf{s}_p^{(i)}, \mathbf{s}_q^{(i)})$$

where $$\ell_{\text{CRF}}(\mathbf{s}_p^{(i)}, \mathbf{s}_q^{(i)}) = \mathbb{1}[\arg\max_k s_{p,k}^{(i)} \neq \arg\max_{k'} s_{q,k'}^{(i)}]$$

Key idea:

- Penalize the assignement of different labels to related pixels
- Typically solved using graph-cuts or mean field approximation

Examples of regularization losses

Conditional random field (CRF):

$$\ell_u(\mathbf{S}^{(i)}) = \sum_{p,q \in \Omega} w_{pq}\ \ell_{\text{CRF}}(\mathbf{s}_p^{(i)}, \mathbf{s}_q^{(i)})$$

where $\ell_{\text{CRF}}(\mathbf{s}_p^{(i)}, \mathbf{s}_q^{(i)}) = \mathbb{1}[\arg\max_k s_{p,k}^{(i)} \neq \arg\max_{k'} s_{q,k'}^{(i)}]$

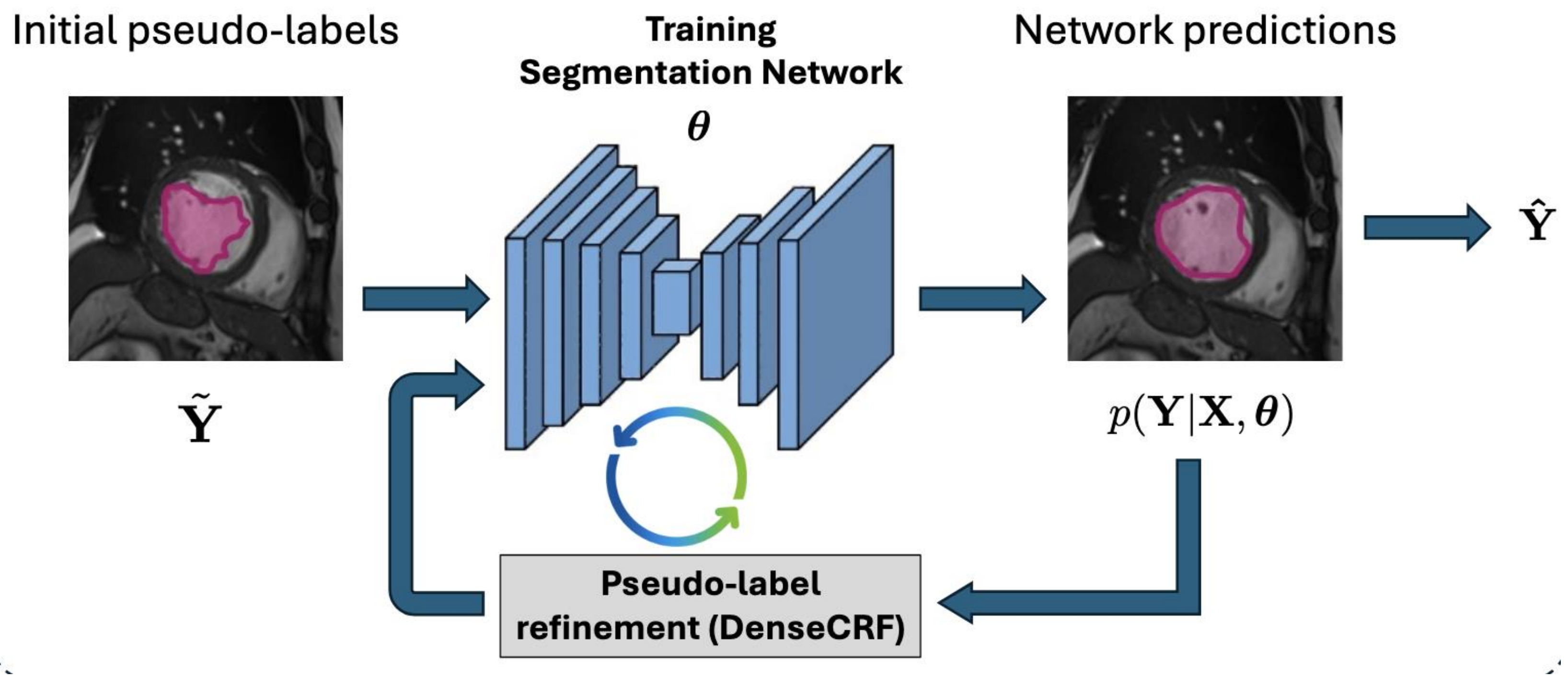
$$\boxed{w_{p,q} = \beta_2 \exp\left(-\frac{\|p-q\|^2}{2\sigma_1^2} - \frac{\|\mathbf{x}_p^{(i)} - \mathbf{x}_q^{(i)}\|^2}{2\sigma_2^2}\right) + \beta_1 \exp\left(-\frac{\|p-q\|^2}{2\sigma_3^2}\right)}$$

Key idea:

- Penalize the assignement of different labels to related pixels
- Typically solved using graph-cuts or mean field approximation

Refining the segmentation

Iterative refinement:



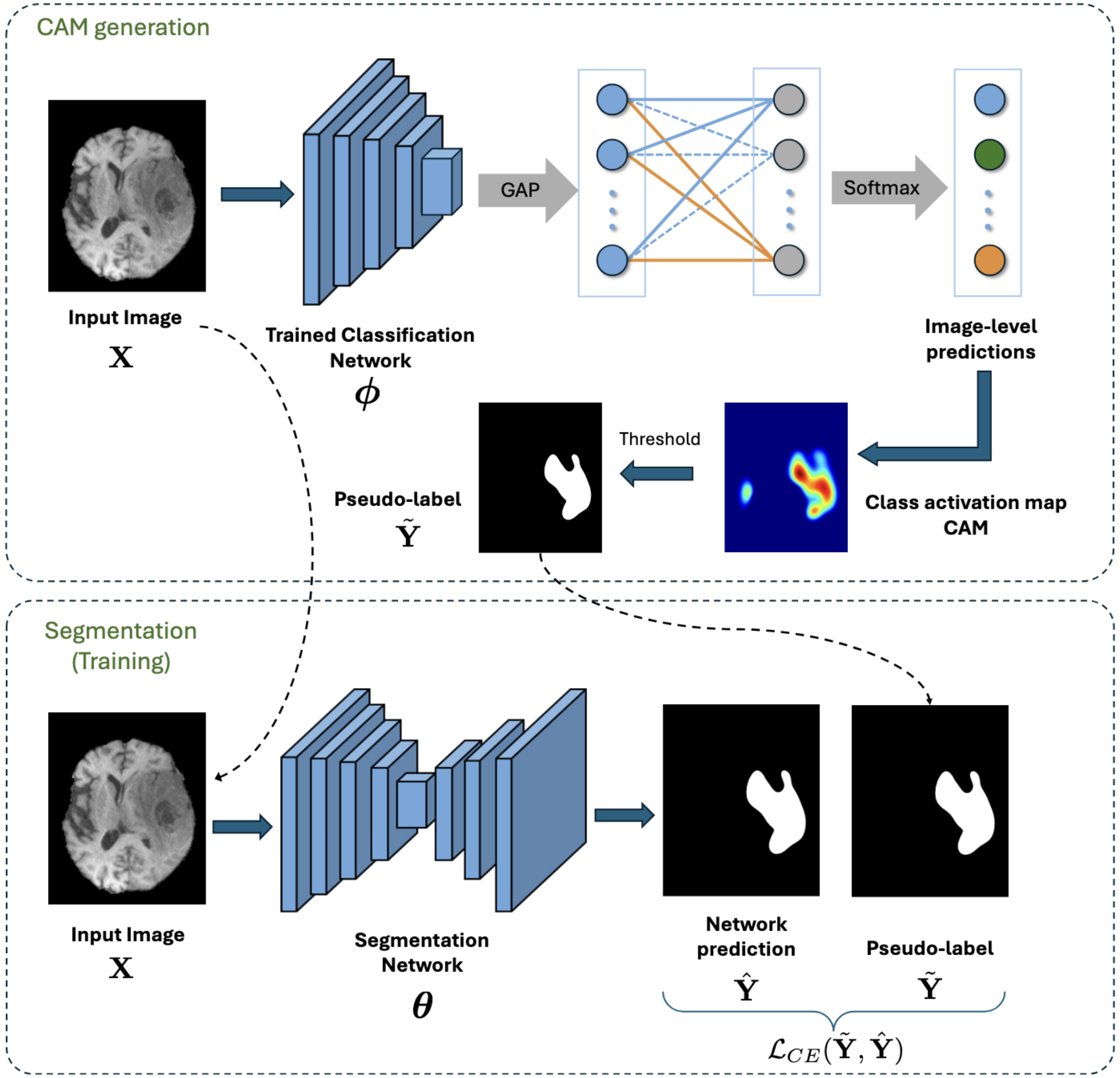
Key idea:

- Use the output of the CRF prediction as pseudo-label for next iteration
- Repeat until convergence

Image from Dolz et al., Less-supervised Segmentation with CNNs, MICCAI Book Series, Springer 2025

Image-level annotations

Using Class Activation Maps (CAMs)



Key idea:

- Train a classifier with image-level annotations
- Use the CAMs of the classifier as pseudo-labels for training the segmentation network

Image from Dolz et al., Less-supervised Segmentation with CNNs, MICCAI Book Series, Springer 2025

Take home messages

- Building large training set of labeled examples is not always possible...
- ... but unlabeled data is often available for ***free***
- Semi-supervised methods (e.g., *adversarial learning, consistency regularization, knowledge distillation*) and self-supervised representation learning can boost performance when labeled data is limited
- Similar approaches can be used to adapt models across different domains (e.g., in *unsupervised domain adaptation or test-time adaptation*)
- Not a silver bullet, can be very challenging at times (e.g., adversarial instability)
- Lots of exciting opportunities for future research !!!

DLMI 2025

Questions ?

Many thanks to

Jose Dolz

Ismail Ben Ayed