Deep Learning MRI Reconstruction

M229 Advanced Topics in MRI

Shu-Fu Shih

5/18/2023

Review: compressed sensing MRI

- 3 main components in compressed sensing MRI
  - The image has a sparse representation in some transform domain
  - The k-space sampling trajectory generates incoherent artifacts in the sparse transform domain
  - It involves a nonlinear reconstruction method

$$argmin_x \quad \| UFx - y \|_2^2 + \lambda \| Wx \|_1$$

Deep learning

- Deep learning is subset of machine learning, which is essentially a neural network with many layers.

- Deep learning network “learns” from a lot of data to perform a variety of tasks in computer vision, natural language processing, bioinformatics…

- The success of deep learning since 2010s
    - Availability of large public datasets
    - Accessibility of GPUs for parallel computing
    - Accessibility of codes and toolboxes for deep neural network training

Deep learning MRI reconstruction

- It's impossible to cover all aspects of deep learning for MRI reconstruction because it's an active and rapid-changing research field.
- In this lecture, we will focus on:
  - Introducing basic components of deep learning networks, especially on ConvNet
  - Providing some insights on why it *might* work
  - Presenting different applications of deep learning in MRI reconstruction

Image reconstruction model

- General image acquisition model: $y = Ax + n$
  - y: the acquired data in the sensor domain (e.g., k-space in MRI)
  - x: the image
  - n: additive noise
  - A: an operator which is modality dependent
    - For computed tomography (CT): A is Radon transform
    - For fully sampled Cartesian MRI: A is Fourier transform
    - For undersampled Cartesian MRI: A includes subsampling and Fourier transform
    - For non-Cartesian MRI: A is non-uniform Fourier transform

Image reconstruction model

- To solve an underdetermined inverse problem (e.g., in the case of undersampled MRI), constrained reconstruction methods have been popular

Image model

$$y = Ax + n$$

Constrained reconstruction optimization problem

$$argmin_x \quad L(x) = \underbrace{F(Ax, y)}_{\text{Consistency with sensor-domain data}} + \lambda \underbrace{\Phi(W, x)}_{\text{Regularization term encoding prior information}}$$

Image reconstruction model

- Deep learning uses information from a large dataset to learn a non-linear mapping.
- In the task for MRI reconstruction from undersampled data:

Non-linear neural network

$$Q = \mathscr{F}(P)$$

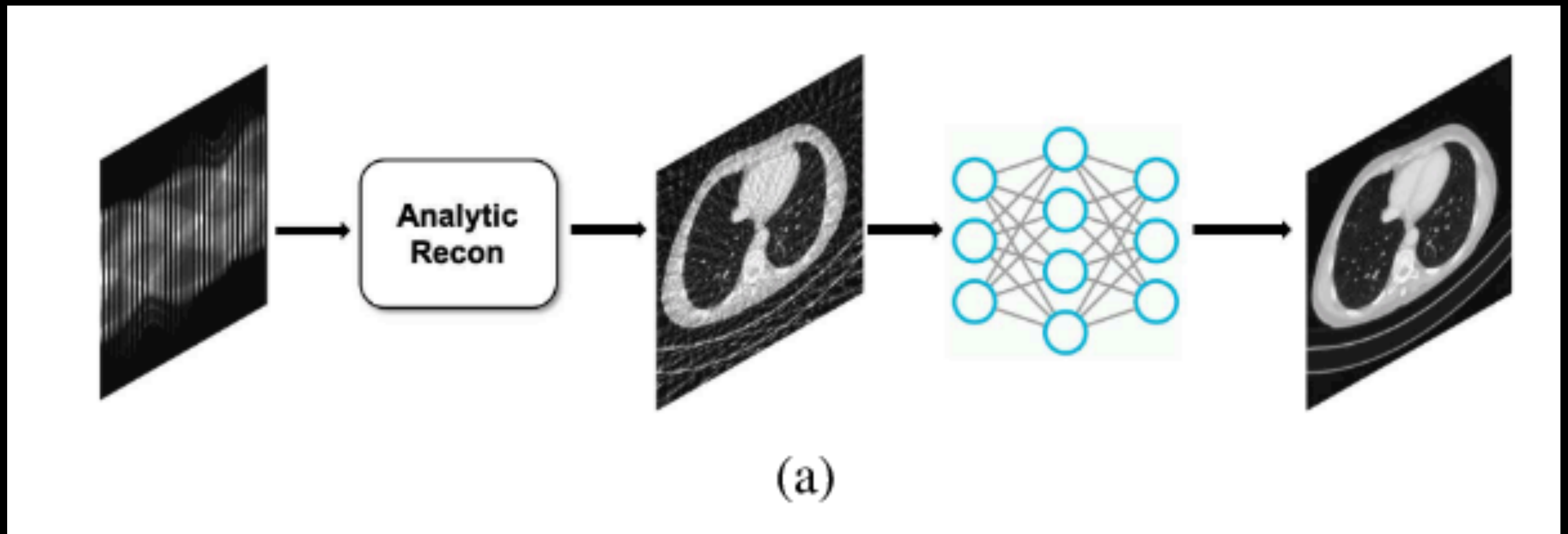
Images with reduced artifacts or fully sampled images/k-space data

Images or k-space data from undersampled measurements

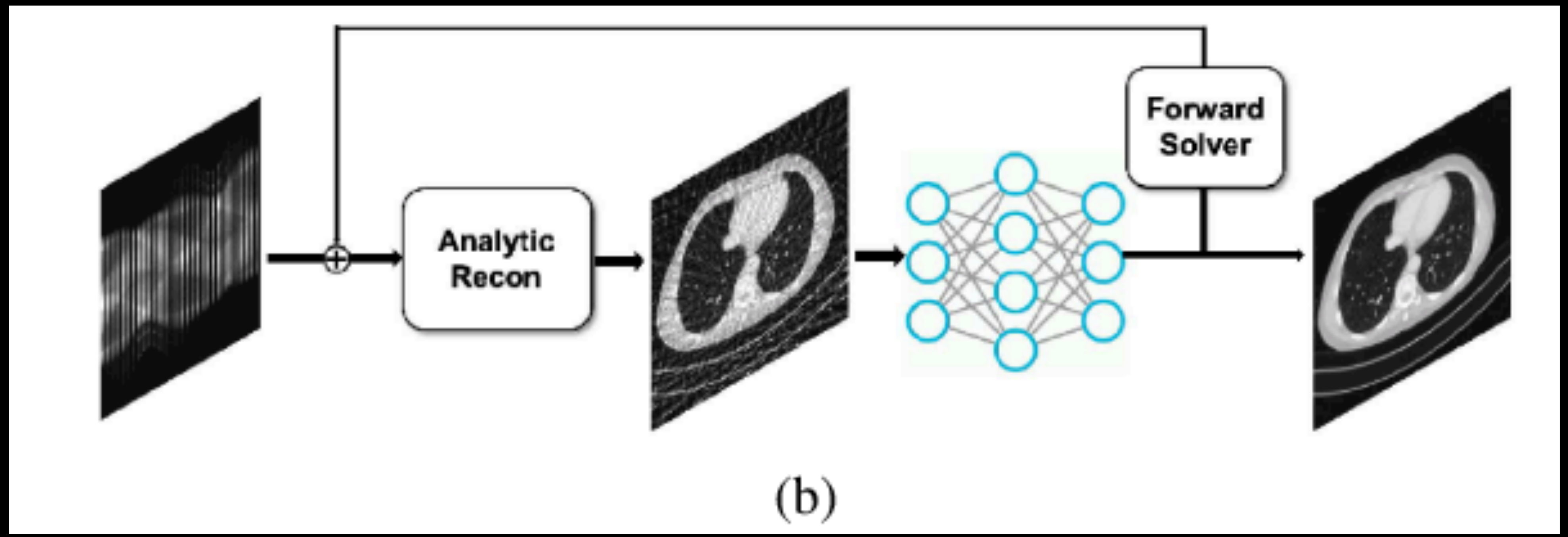
Deep learning medical imaging reconstruction

- Different realizations:

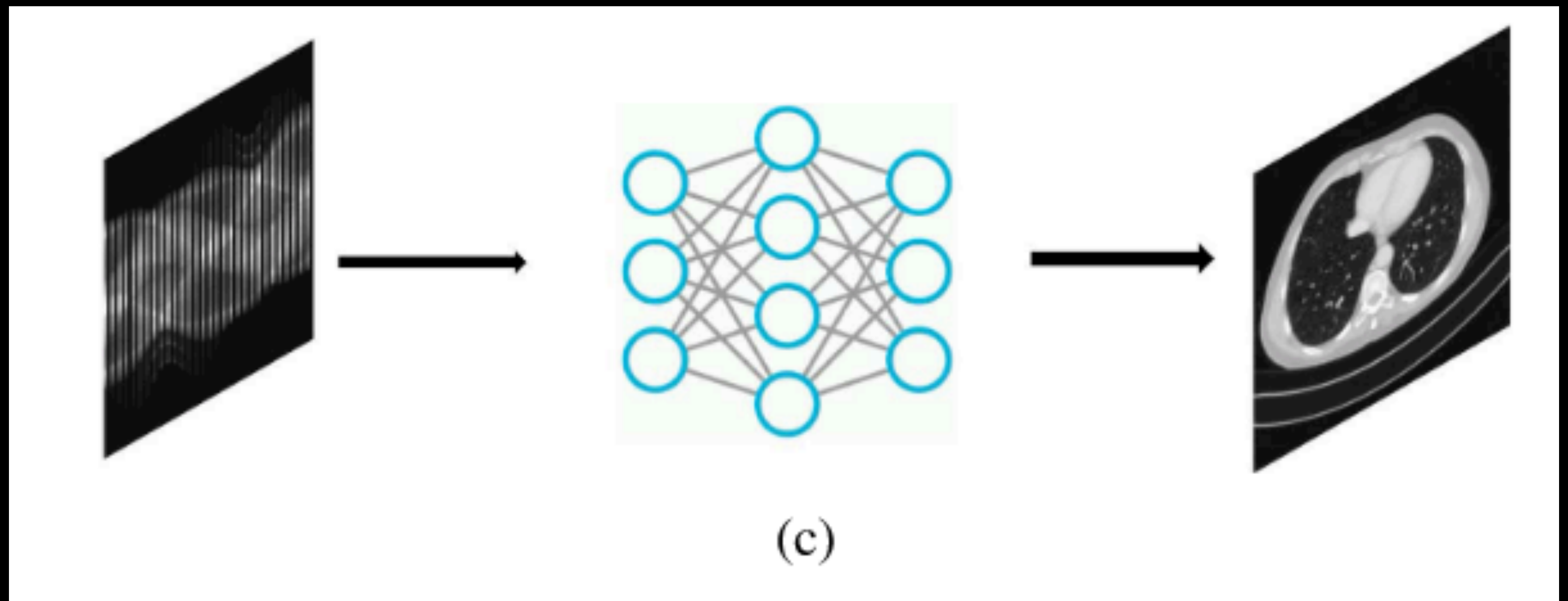
Image-domain learning



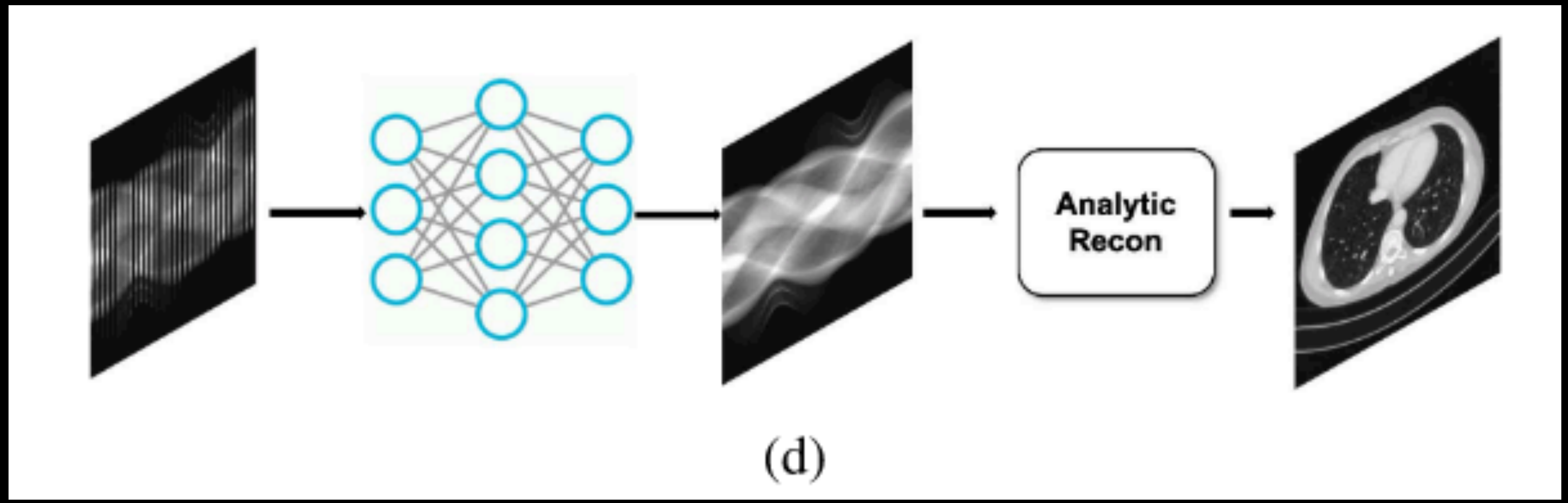
Hybrid-domain learning



Mapping between sensor domain
and image domain



Sensor-domain learning



(Figures from: Ravishankar et al., Proceedings of the IEEE 2020)

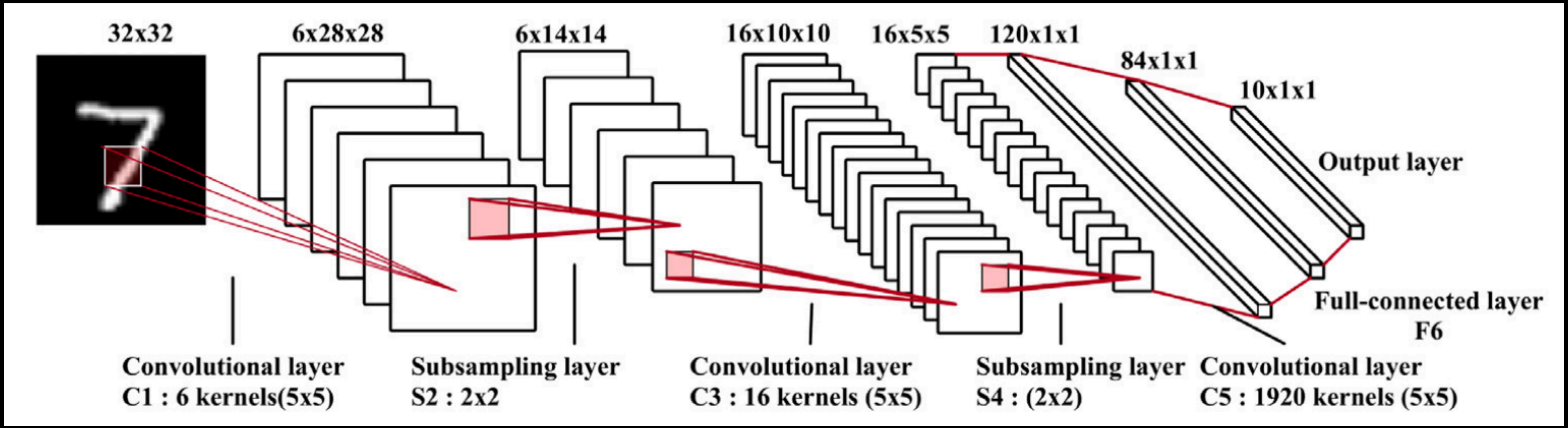
Convolution Neural Networks (CNN or ConvNet)

- ConvNet is one of the most popular deep learning network for imaging tasks

- We will introduce several key components in ConvNet and show how it can be trained
  - Convolution layer
  - Pooling layer
  - Activation function
  - Loss function
  - Optimizer
  - Regularization
  - Batch normalization
  - …

Where it all started…

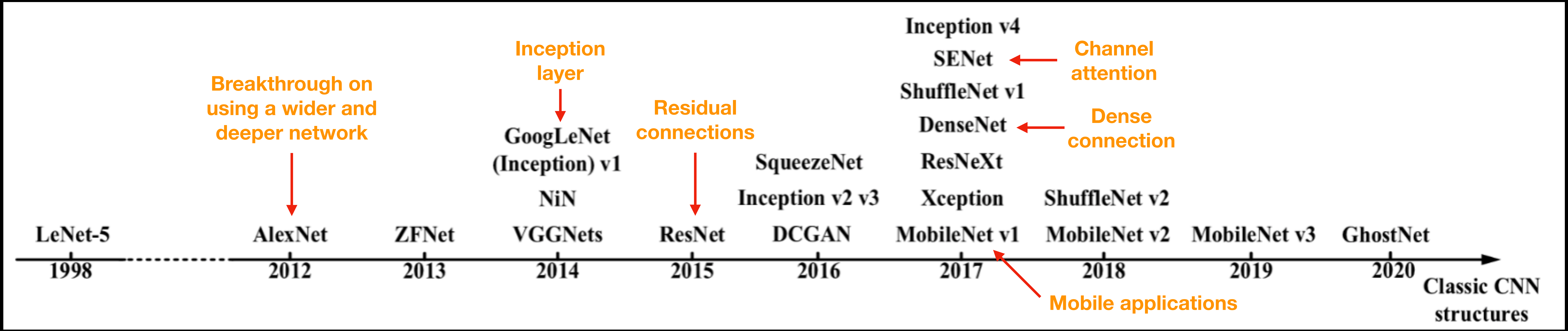
- LeNet-5[1]: one of the very first ConvNet architectures with back-propagation for handwritten digit recognition



[1] LeCun et al., Proceedings of the IEEE, 1998

(Figure from: Gu et al., Pattern Recognition, 2018)

A glimpse of popular ConvNet models

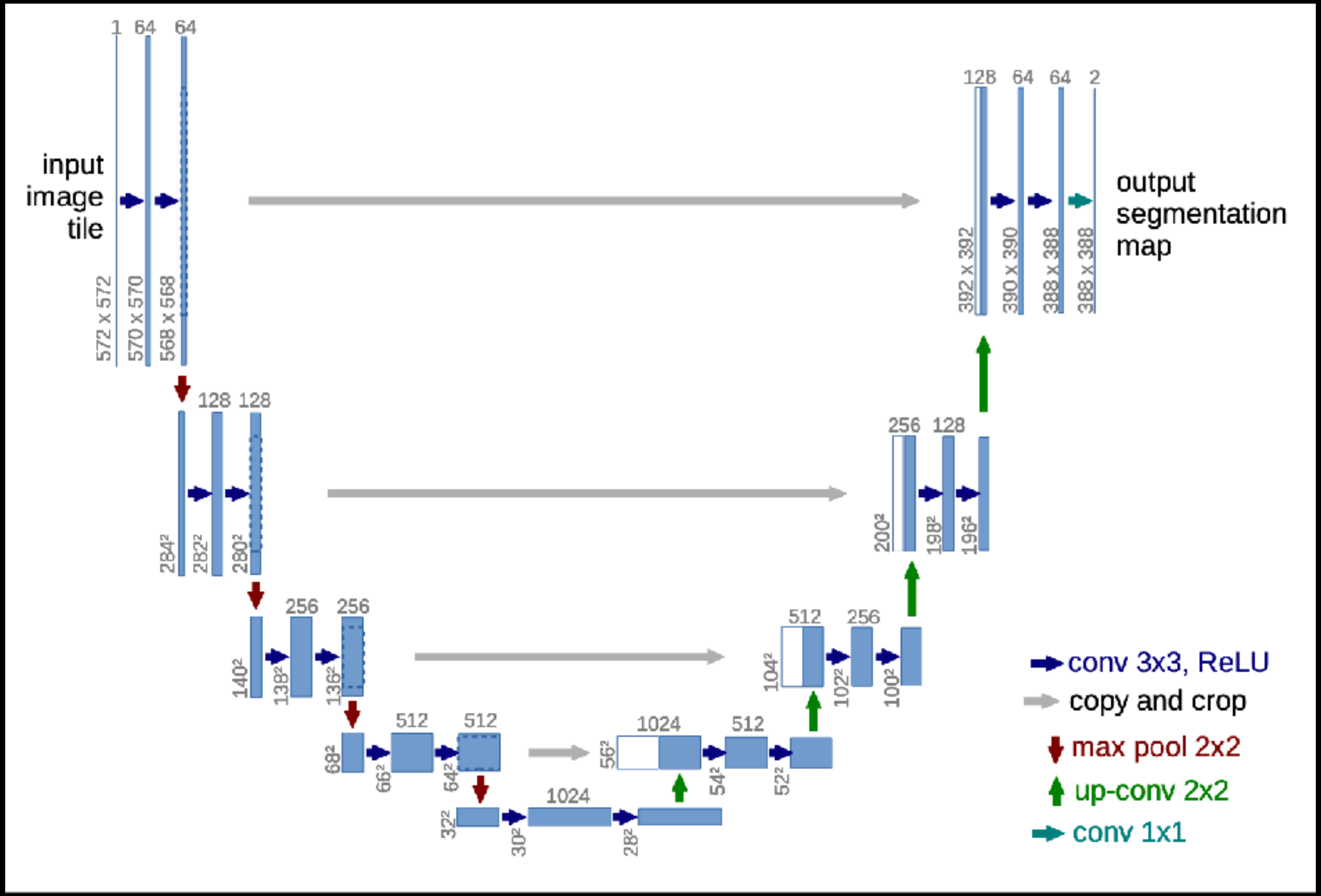


* Many of these ConvNet were first used in natural images (not medical images) and in a variety of tasks (e.g., classification, segmentation…)

(Figure from: Li et al., IEEE Trans Neural Netw Learn Syst 2022)

Popular ConvNet: U-Net

- The original U-Net was designed for medical image segmentation.
- It has been modified and applied in many DL-based MRI reconstruction tasks.



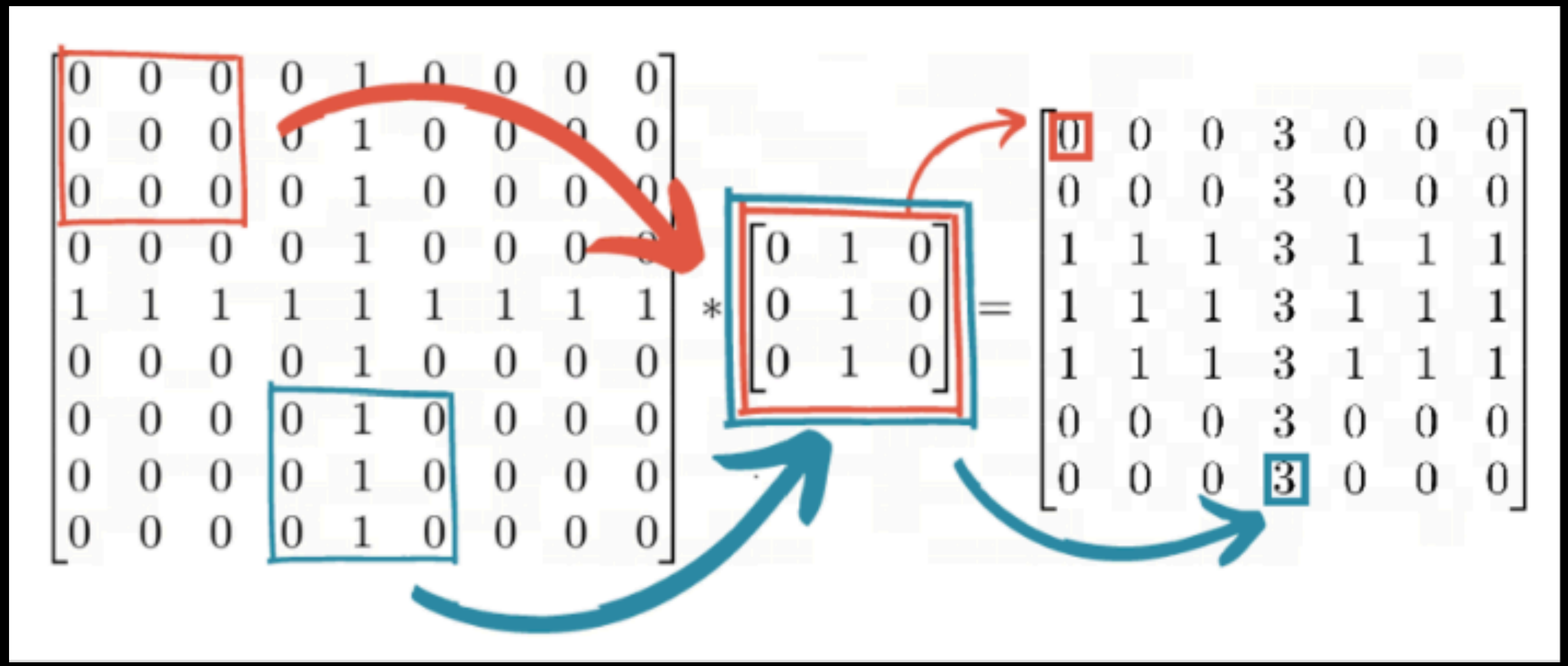
- Convolution at different levels
- Pooling layers
- Contracting and expansive paths
- Skipped connections

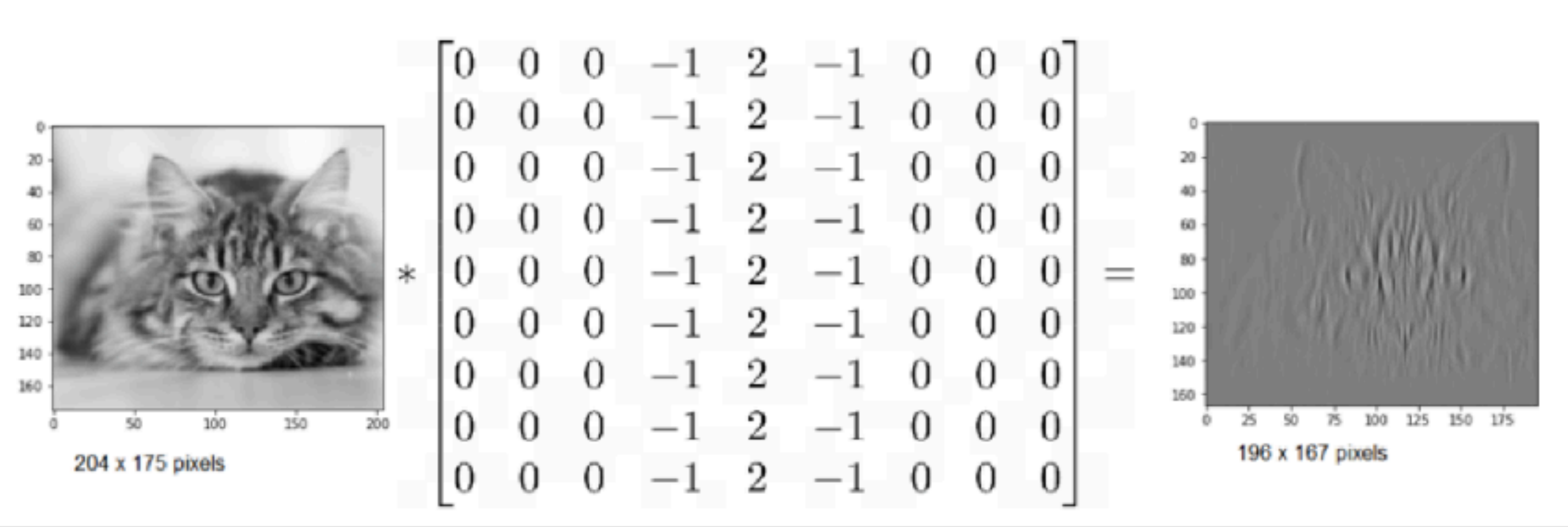
[1] Ronneberger et al., MICCAI, 2015

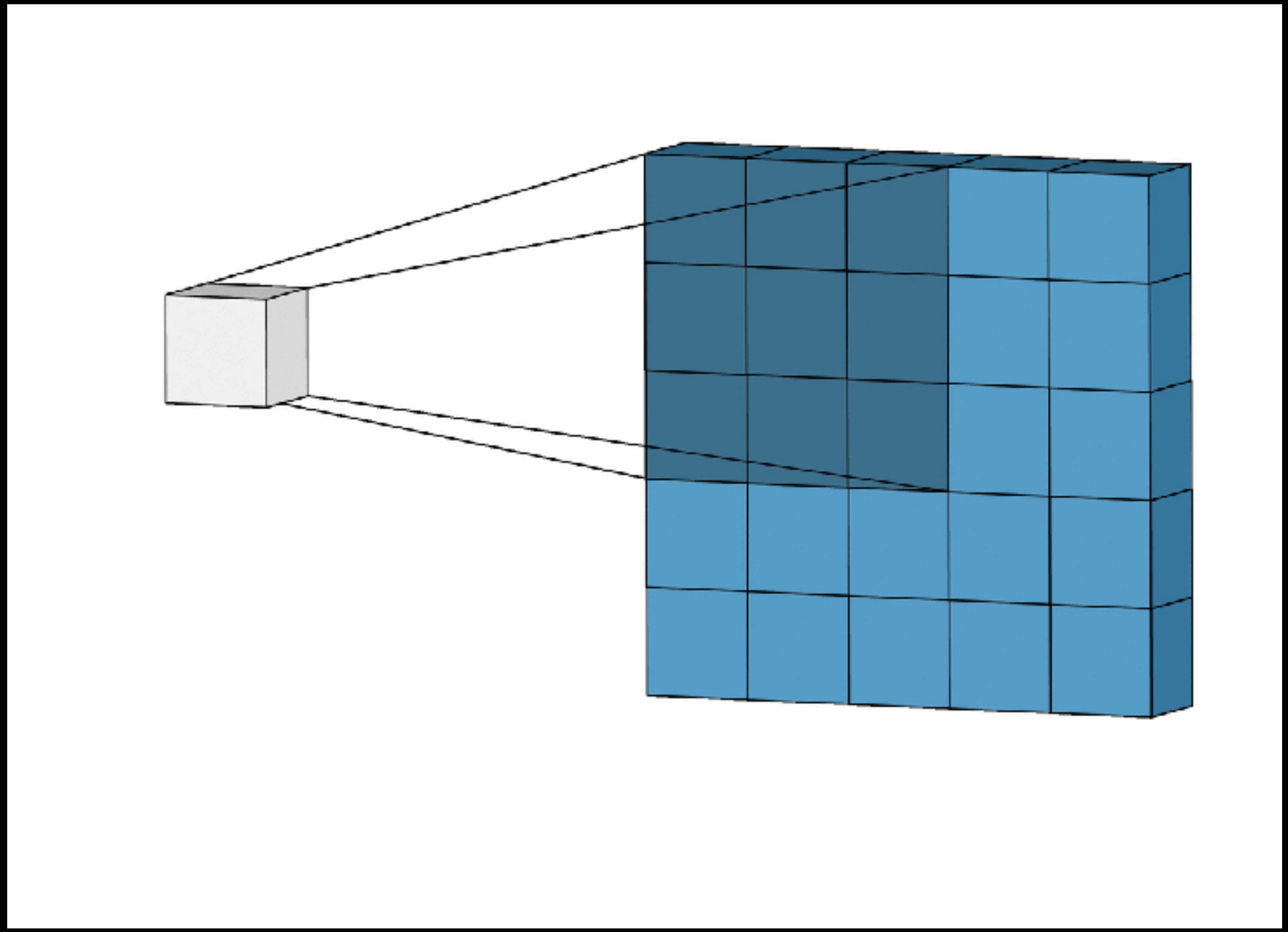
(Figure from: Ronneberger et al., MICCAI, 2015)

Convolutional layer

- Convolution operation: use a shared kernel to convolve with the entire image







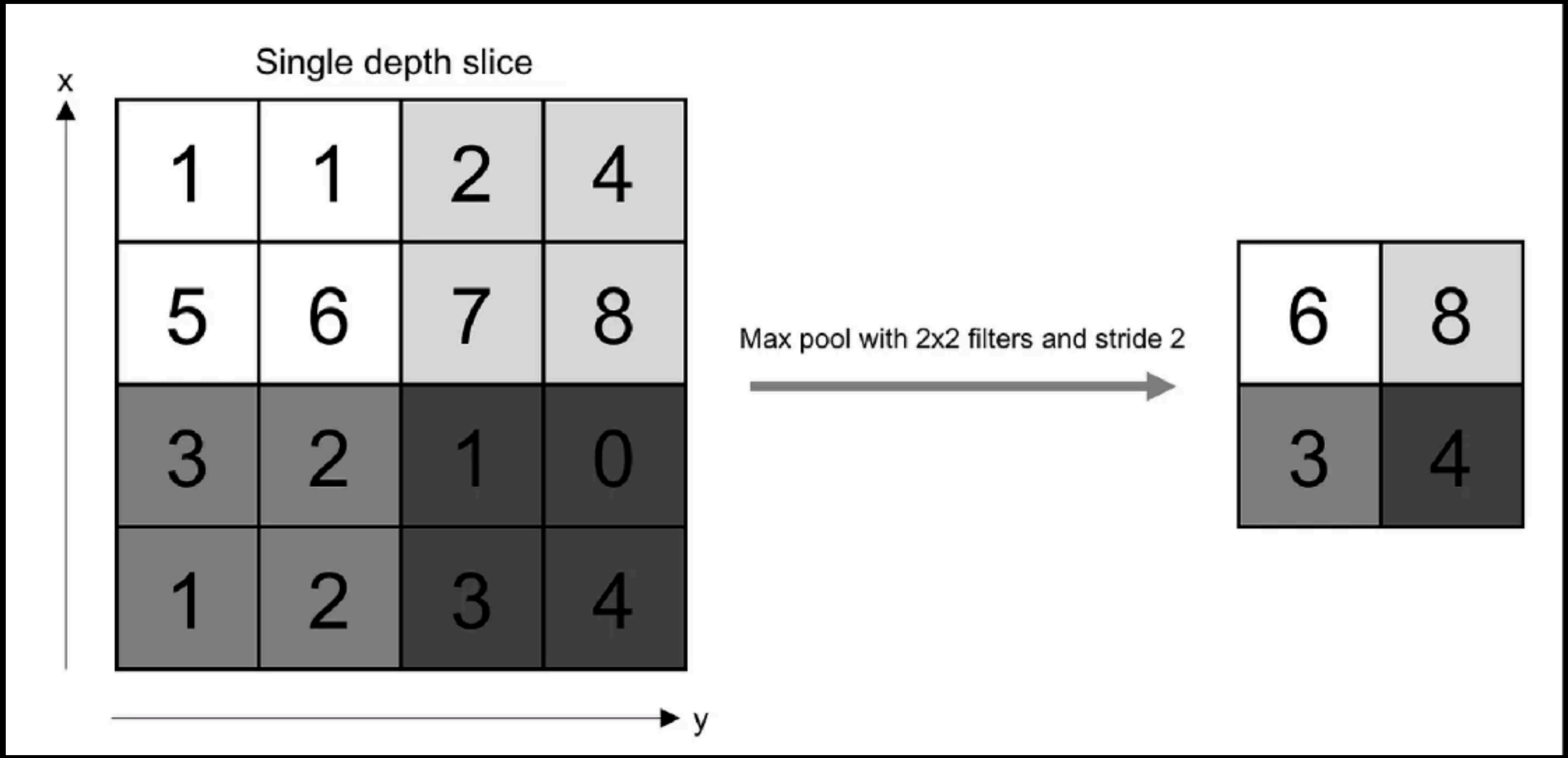
Figures from: https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network

Convolutional layer

- Motivation of using a convolutional layer
- (1) Sparse interaction
    - Each pixel interacts with the kernel instead of all the other pixels.
- (2) Translational invariance
    - Some features are shared across the entire image.
    - The features do not change if the input is shifted.

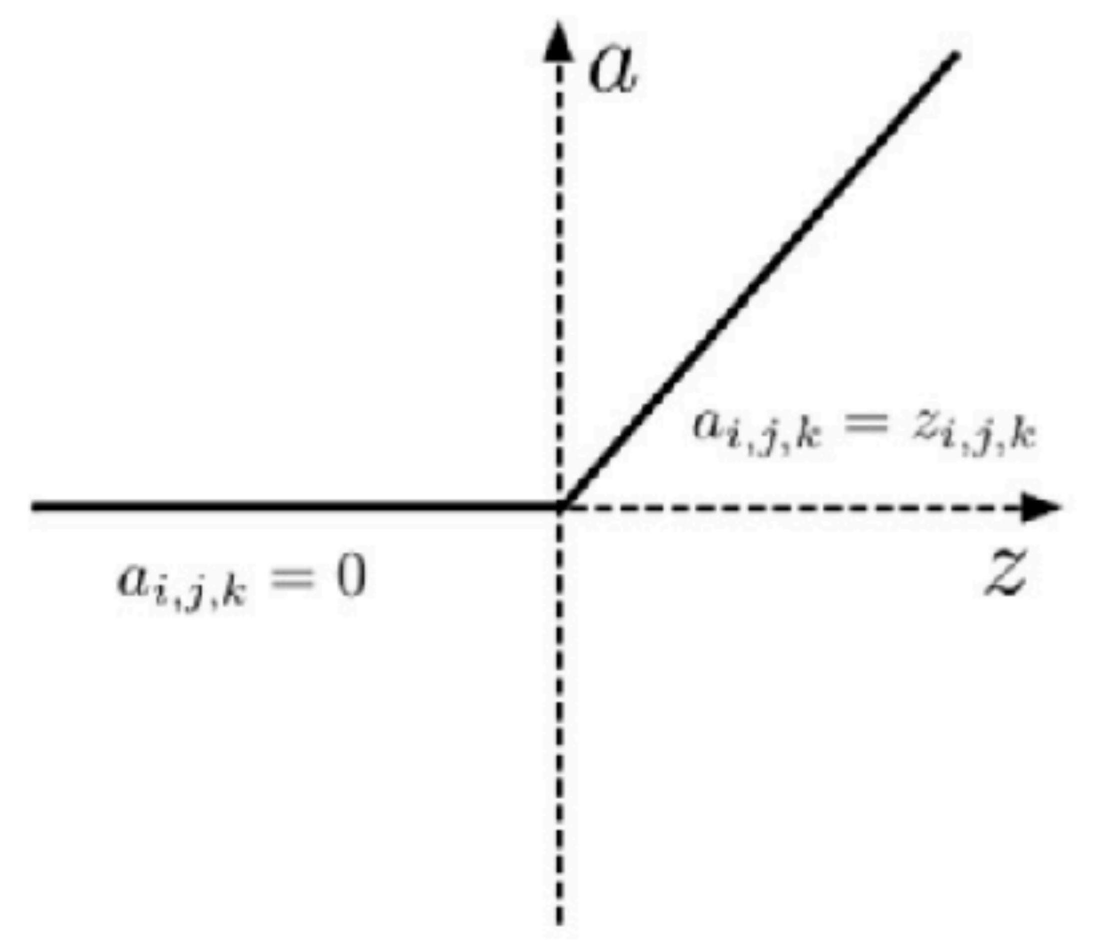
Pooling layer

- Generate a summary of statistics with a reduced number of weights
  - Stride: the number of pixel shift for the next pooling operation



Activation function

- Convolution operation is linear. A stack of convolutional layers only generates a linear mapping process.
- Activation functions are used to introduce non-linearity to the network.
- ReLU (rectified linear unit): $f(a) = max(0, a)$

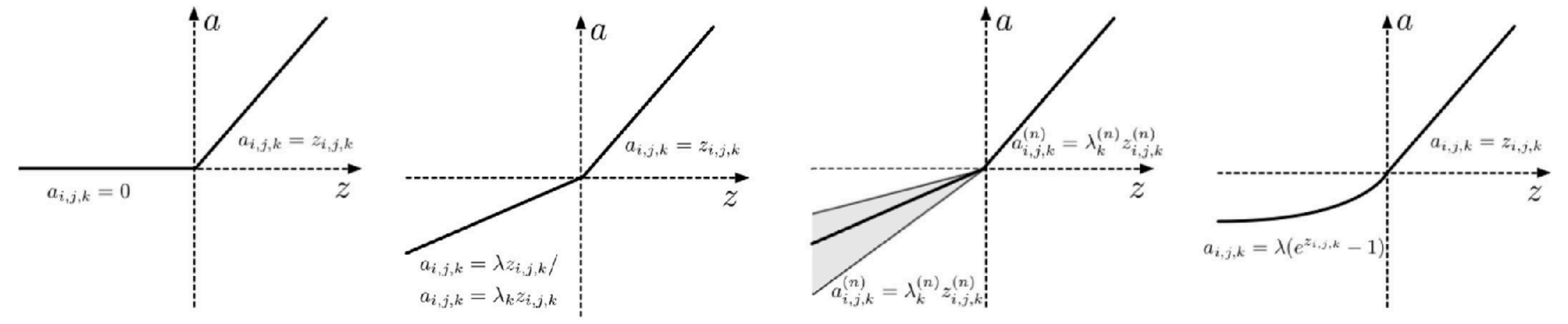


(a) ReLU

(Figures from: Gu et al., Pattern Recognition, 2018)

Improvements on activation functions

- ReLU has zero gradient when the node is not active
  - Different activation functions have been proposed to alleviate the problem



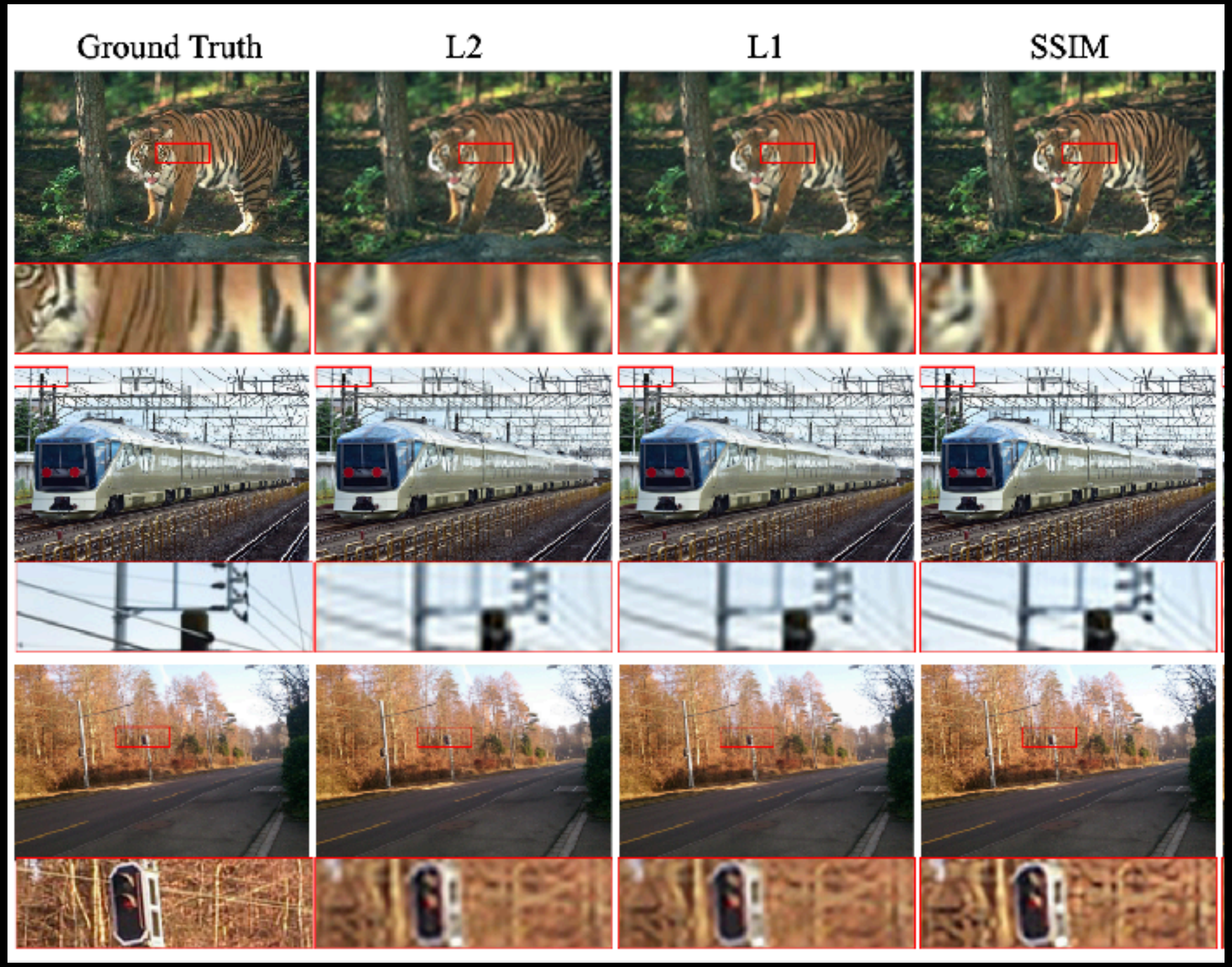
(a) ReLU (b) LReLU/PReLU (c) RReLU (d) ELU

(Figures from: Gu et al., Pattern Recognition, 2018)

Loss function

- We need an objective criteria to tell the network how well it performs.
- The overall network is trained to minimize the loss function.

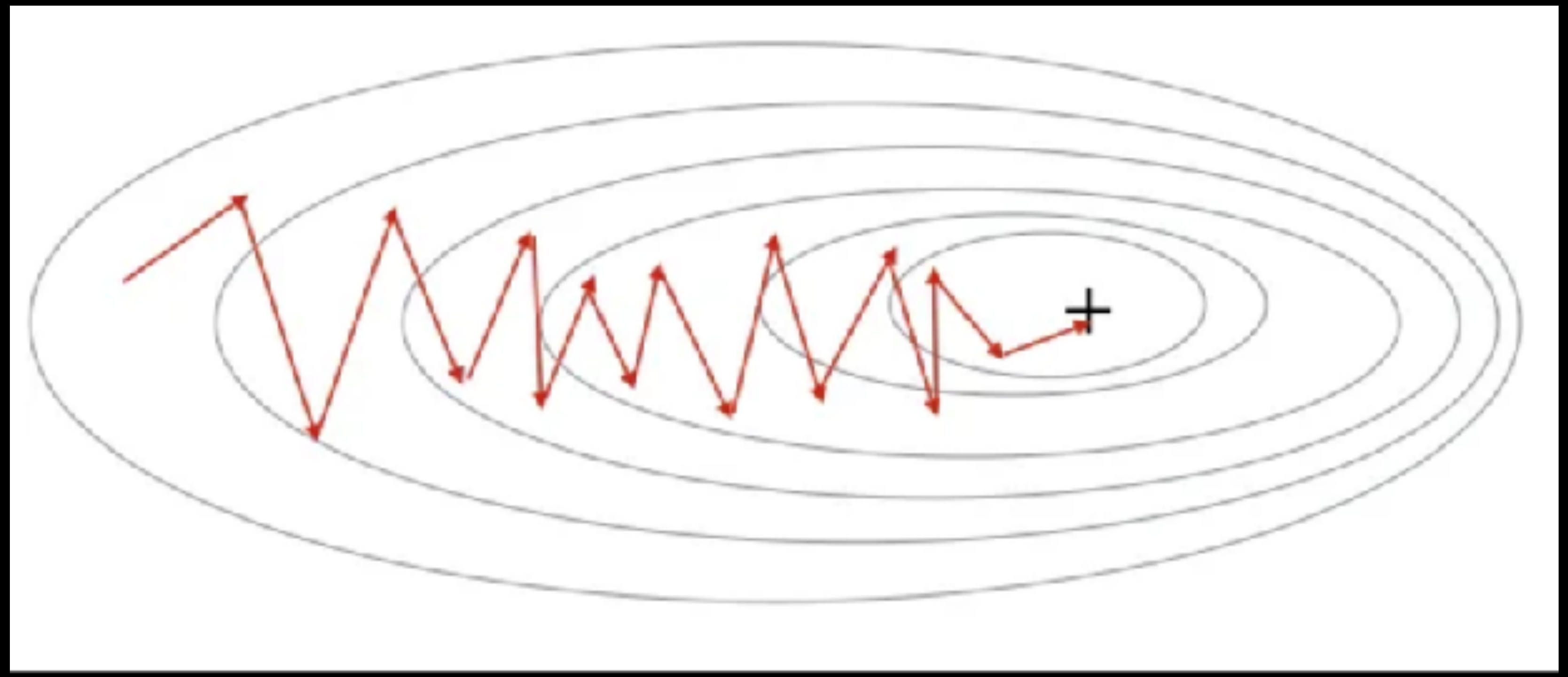
- Loss functions for image reconstruction:
  - MSE loss / L2 loss
  - L1 loss
  - SSIM (structural similarity index measure) loss
  - perceptual loss
  - GAN (generative adversarial network) loss
  - …



(Figure from: Mustafa et al., WACV, 2022)

Optimizer

- Algorithms used to update network parameters for loss minimization
  - Gradient descent
  - Stochastic gradient descent
    - Replace the actual gradient calculation from the entire dataset by using a randomly selected subset
    - “Batch size” can be used to refer to the number of training samples in one forward/backward pass

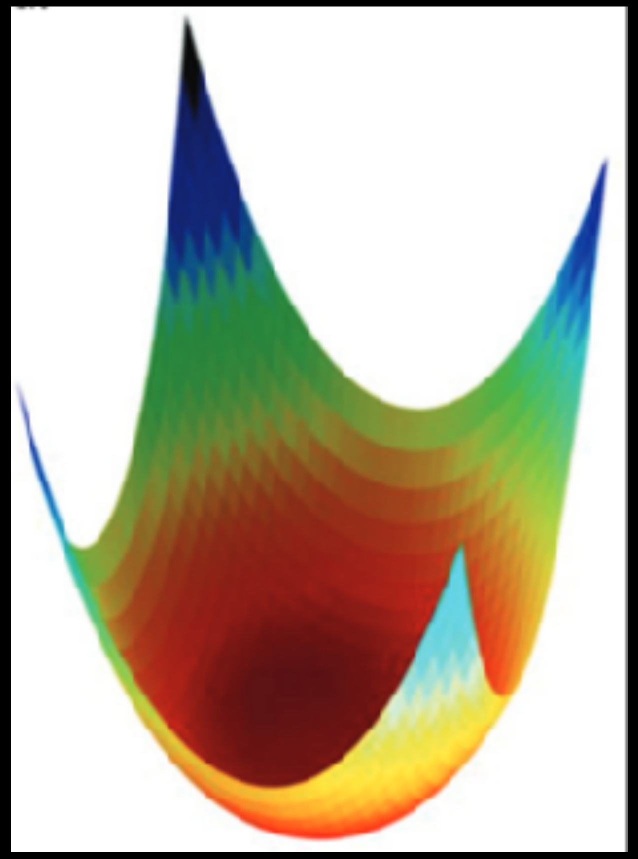


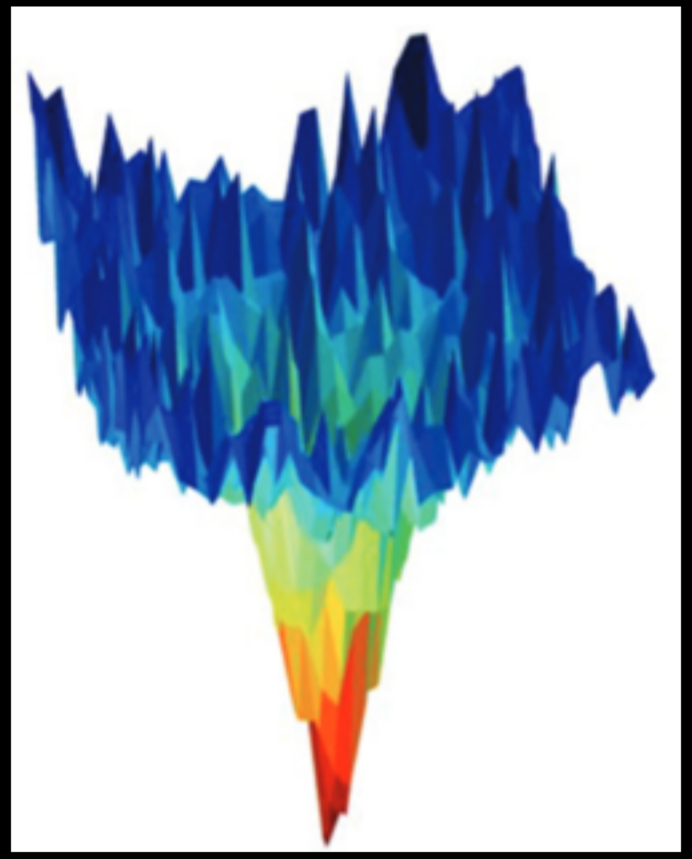
Stochastic gradient descent

(Figure from: https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0)

Optimizer

- To avoid local minimum problems, there are more adaptive optimizers that incorporate a “momentum” idea that use previous gradient information
  - Adagrad
  - RMSProp
  - Adam
  - …



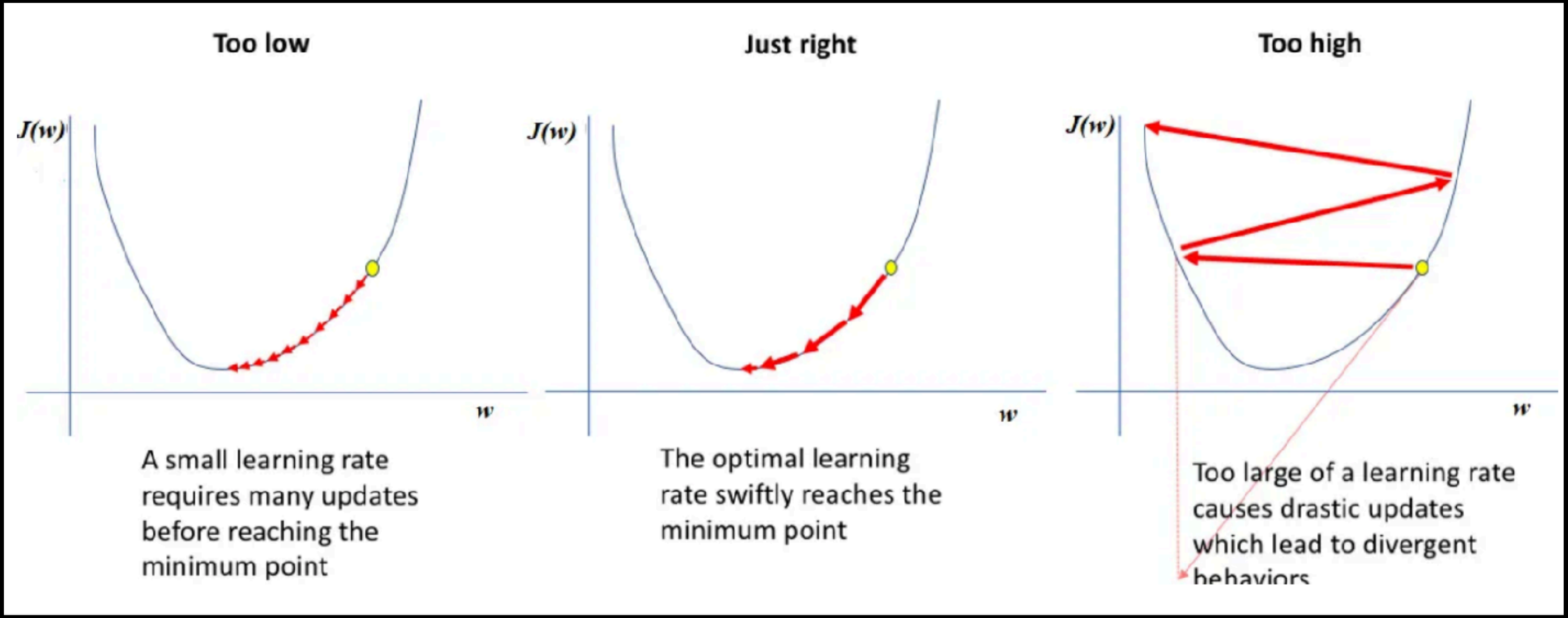


- Luckily, there are many optimizers already implemented in popular deep learning frameworks (PyTorch, TensorFlow…)

(Figures from: Cheng et al., RSNA, 2021)

Optimizer

- Find a suitable learning rate



(Figure from: https://towardsdatascience.com/hyper-parameter-tuning-techniques-in-deep-learning-4dad592c63c8)

Back-propagation

- Once we know about the gradient, back-propagation is usually used as an efficient way to update the network's trainable parameters.

Back-propagation

One layer:

$$\underbrace{\phi}_{\text{Nonlinear function}}(\underbrace{w}_{\text{Weight}}\underbrace{P}_{\text{Input image}} + \underbrace{b}_{\text{Bias}})$$

$$Q = \phi(wP + b)$$

Network with deep layers:

$$Q = \mathcal{F}(P) = \phi(w_n \ldots \underbrace{\phi(w_2 \underbrace{\phi(w_1 P + b_1)}_{\text{First layer}} + b_2)}_{\text{Second layer}} \ldots + b_n)$$

Using chain rule
To calculate derivatives

$$Q = \mathcal{F}(g(P)) \longrightarrow \frac{\partial Q}{\partial P} = \frac{\partial \mathcal{F}(P)}{\partial P} = \mathcal{F}'(g(x)) \,.\, g'(x)$$

- Luckily, back-propagation can be done easily using popular deep learning frameworks (PyTorch, TensorFlow…)

Regularization

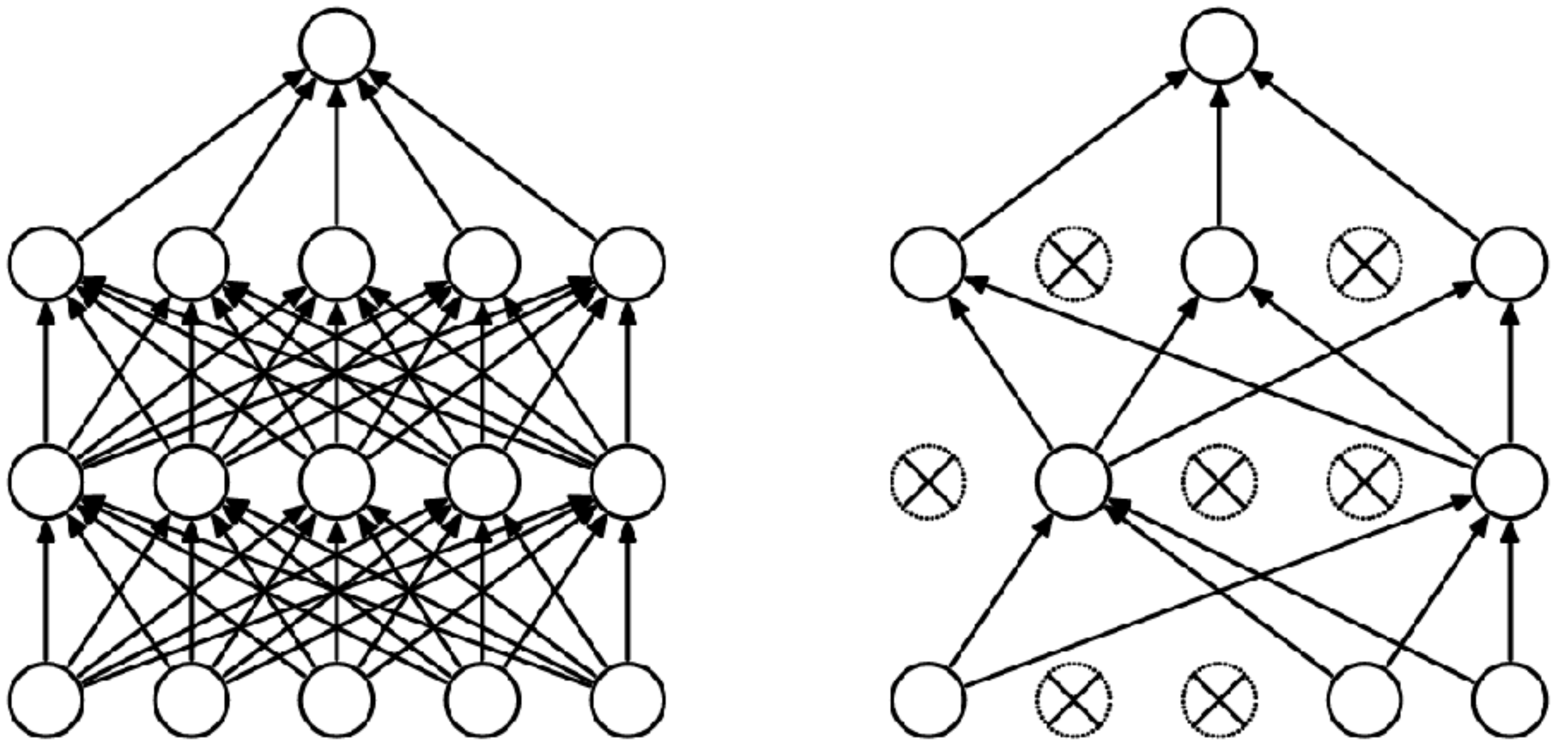
- Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error[1].

- Examples:
  - Include prior knowledge
  - Apply some constraints on the parameters in the loss function
  - Data augmentation: image flipping, rotation…
  - Dropout
  - …

[1] Goodfellow et al., *Deep learning*. MIT press, 2016

Regularization

- Dropout[1]
  - Randomly “turn off” some of the weights during the training process.



(a) Standard Neural Net (b) After applying dropout.

[1] Srivastava et al., JMLR, 2014

Batch normalization

- Internal covariance shift[1]
  - The distribution of the inputs in each layer changes as learning occurs in previous layers.
- Batch normalization[1] normalizes output of the previous layer by subtracting the batch mean, and then dividing by the batch's standard deviation (i.e., normalizing the previous output)

[1] Ioffe et al., PMLR, 2015

Data stratification

- A proper data stratification ensures that training and evaluation data is representative of the distributions in the population.

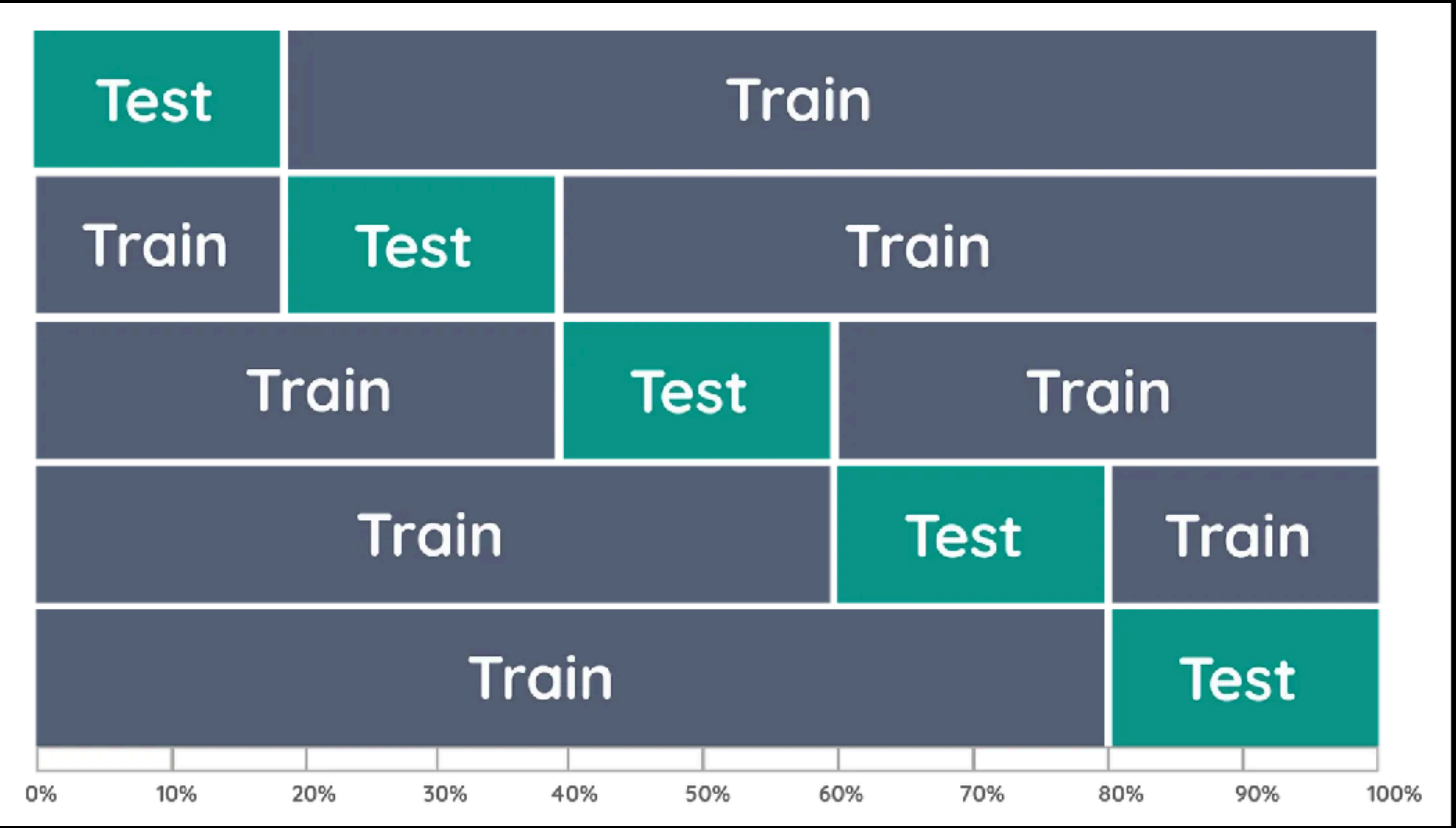
- Things to consider in MRI applications:
  - Subject demographics (sex, age,…)
  - Patients/Healthy volunteers
  - Different diseases
  - Sequence acquisition parameters
  - …

Validation

- Different validation methods
  - Train/test split
  - k-Fold cross validation
  - Leave-one-out cross validation
  - …

Validation

- k-fold cross validation

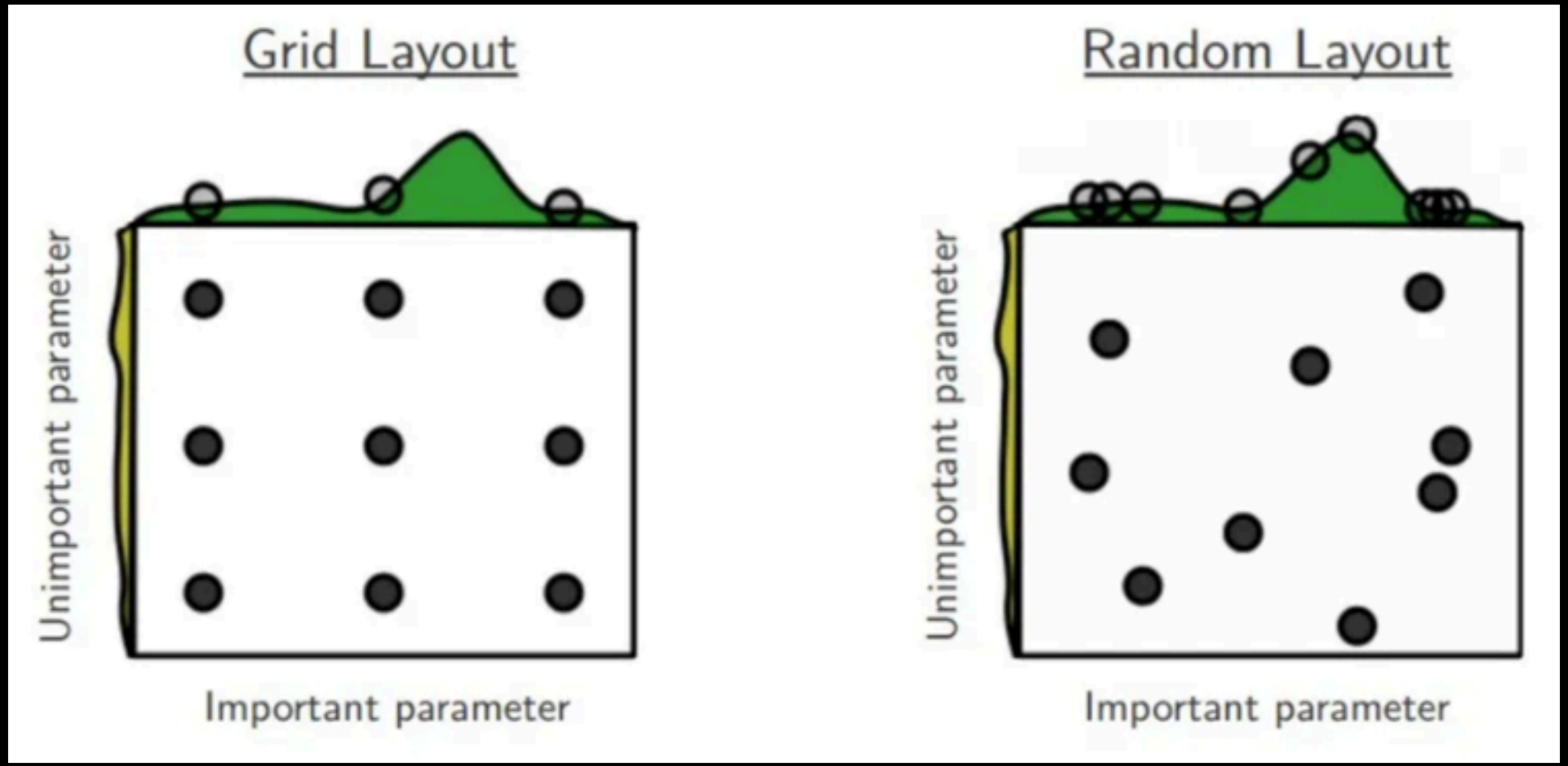


(Figure from: https://towardsdatascience.com/validating-your-machine-learning-model-25b4c8643fb7)

Hyperparameter tuning

- There are many hyperparameters in deep learning networks
  - Learning rate
  - Batch size
  - Architecture design: number of layers, numbers of channels
  - …

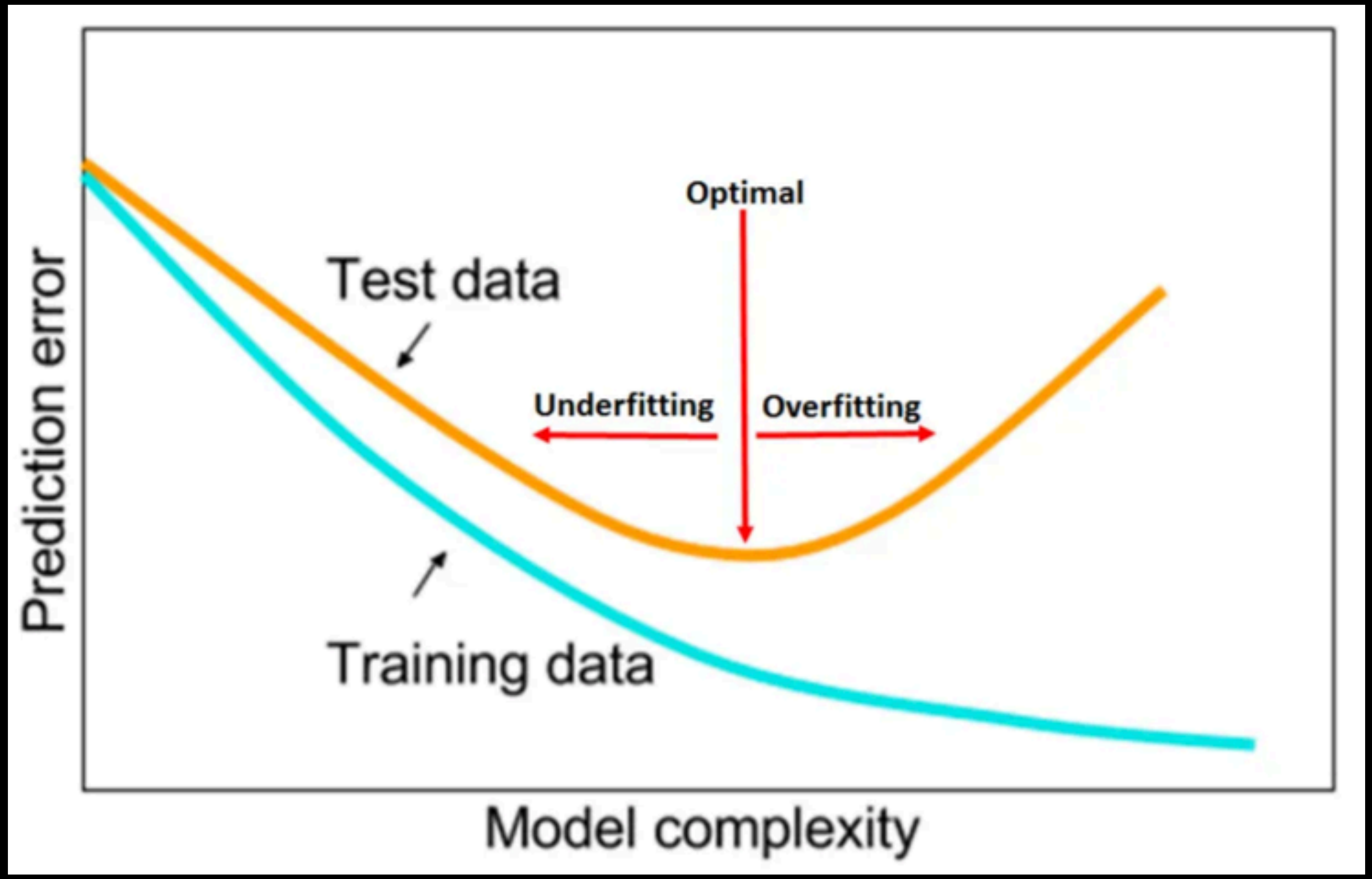
- Approaches for hyperparameter tuning
  - Grid search
  - Random search



https://towardsdatascience.com/hyper-parameter-tuning-techniques-in-deep-learning-4dad592c63c8

Hyperparameter tuning

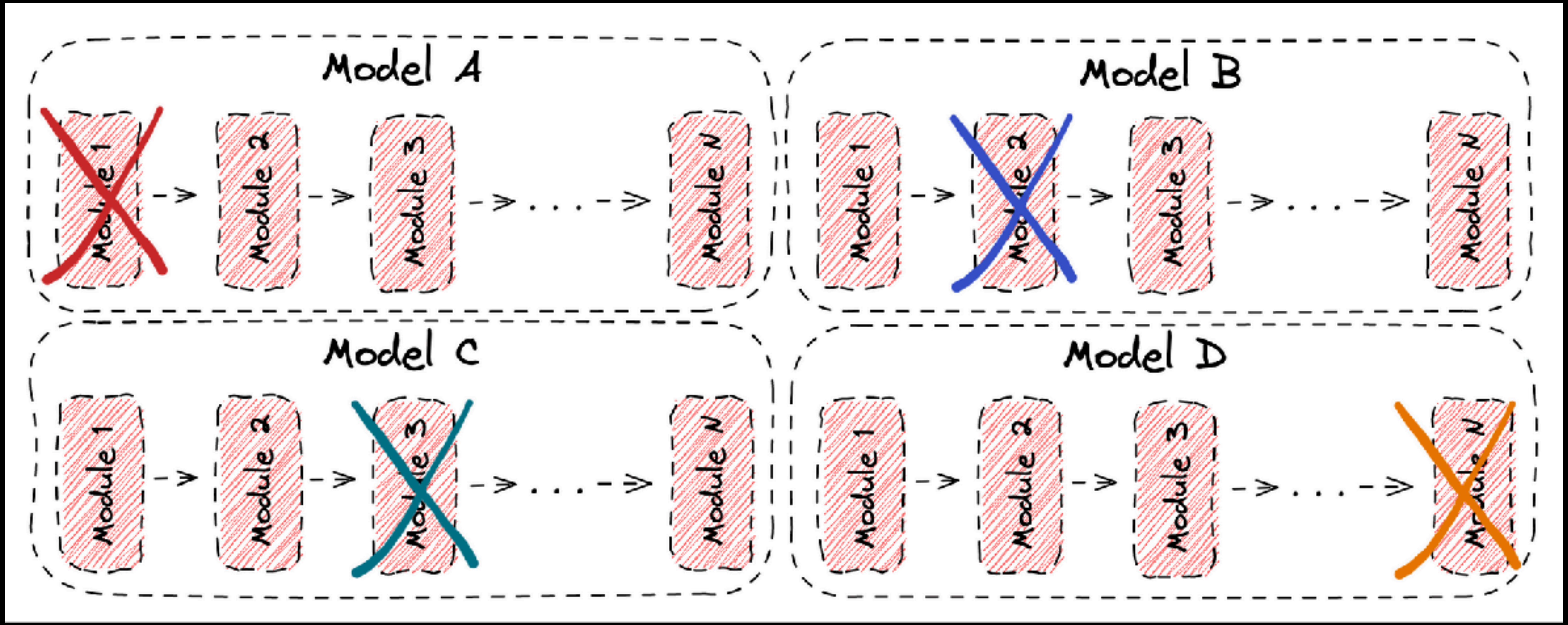
- Monitor validation loss for hyperparameter tuning
- Pay attention to signs of underfitting and overfitting



https://towardsdatascience.com/hyper-parameter-tuning-techniques-in-deep-learning-4dad592c63c8

Ablation study

- Ablation study investigates the performance of a neural network by removing one or several components at a time to understand the contribution from each component to the entire network.



(Figure from: https://www.baeldung.com/cs/ml-ablation-study)

Image quality evaluation

- Quantitative image quality metrics
  - NRMSE, PSNR, SSIM…
- Radiology scoring
  - Experienced radiologists review and rate the image quality
- Statistical analysis

Deep learning-based MRI reconstruction

- Now we will show different deep learning-based MRI reconstruction methods
- We will focus on:
  - What kind of *problem* does it want to solve?
  - What kind of *approach* does it propose?

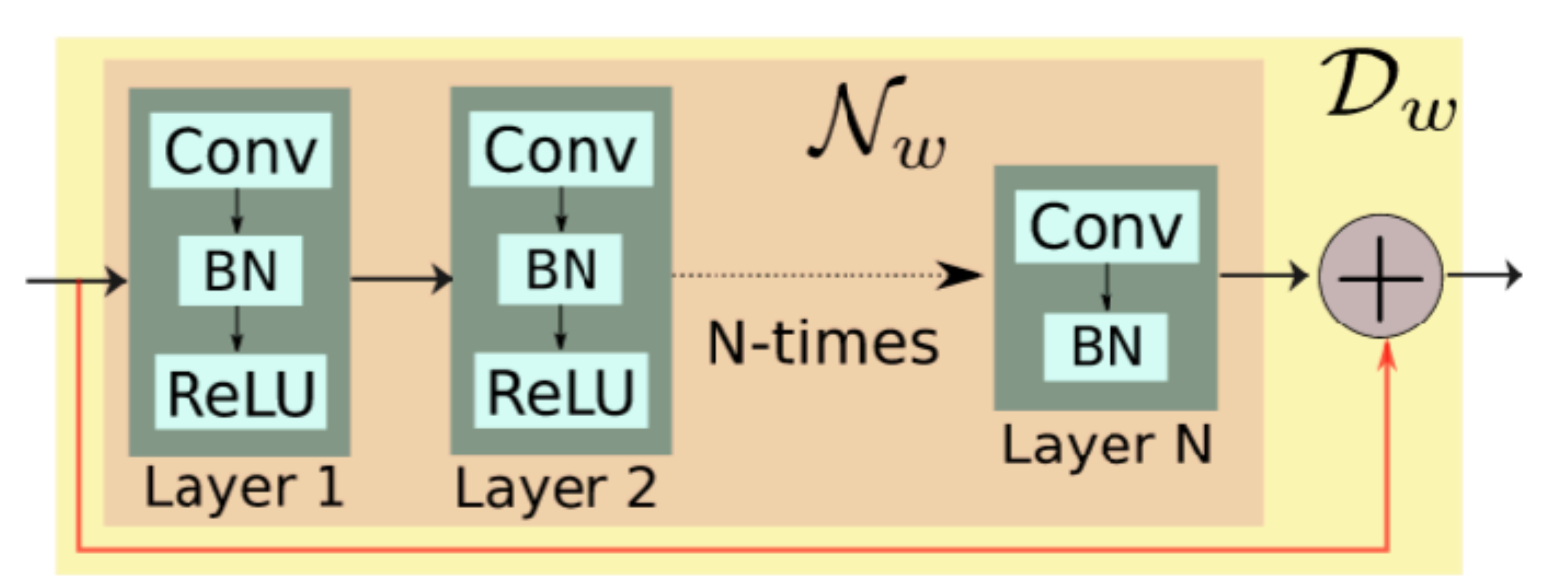
(1) MoDL

- MoDL (Model-based Deep Learning architecture for inverse problem)
  - Replace sparsity constraints (in CS formulation) with a deep learning network

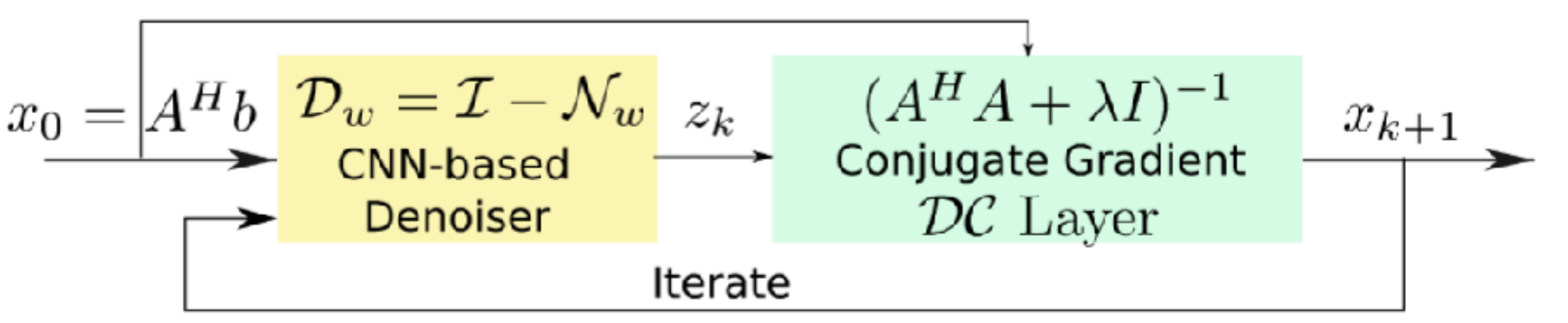
Formulate as an optimization problem

$$x_{recon} = argmin_x \parallel UFx - y \parallel_2^2 + \lambda \parallel x - ConvNet(x) \parallel_2^2$$

An unrolled network with two main blocks
(1) A ConvNet to reduce artifacts / improve image quality
(2) A data consistency layer for k-space data consistency



(a) The Residual learning based denoiser

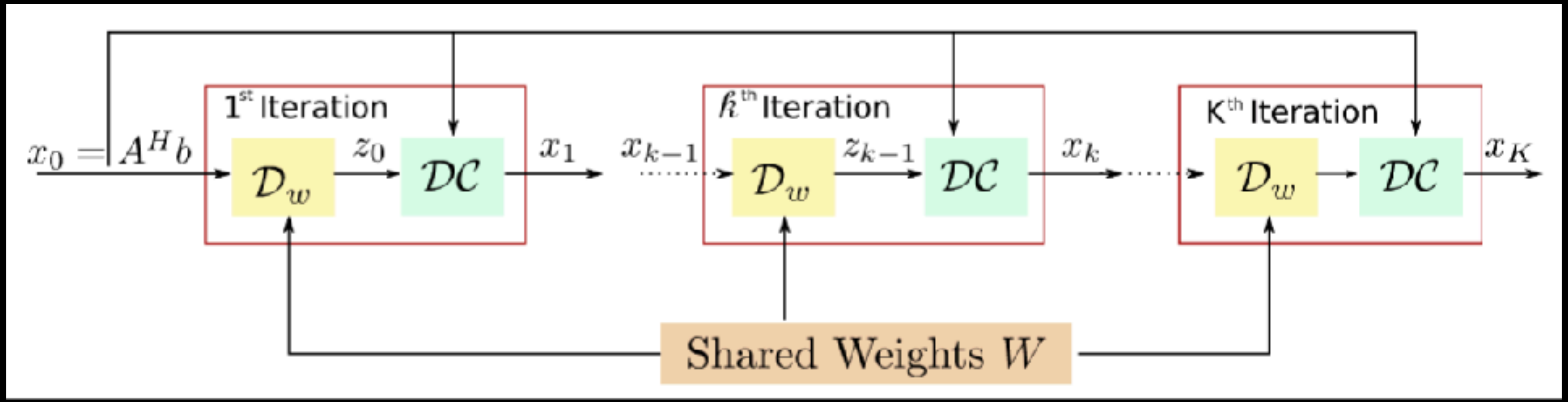


(b) Proposed Model-based Deep Learning (MoDL) architecture

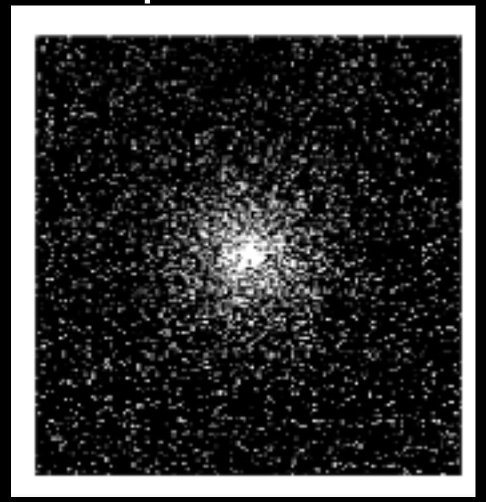
(Figures from: Aggarwal et al., IEEE TMI 2019)

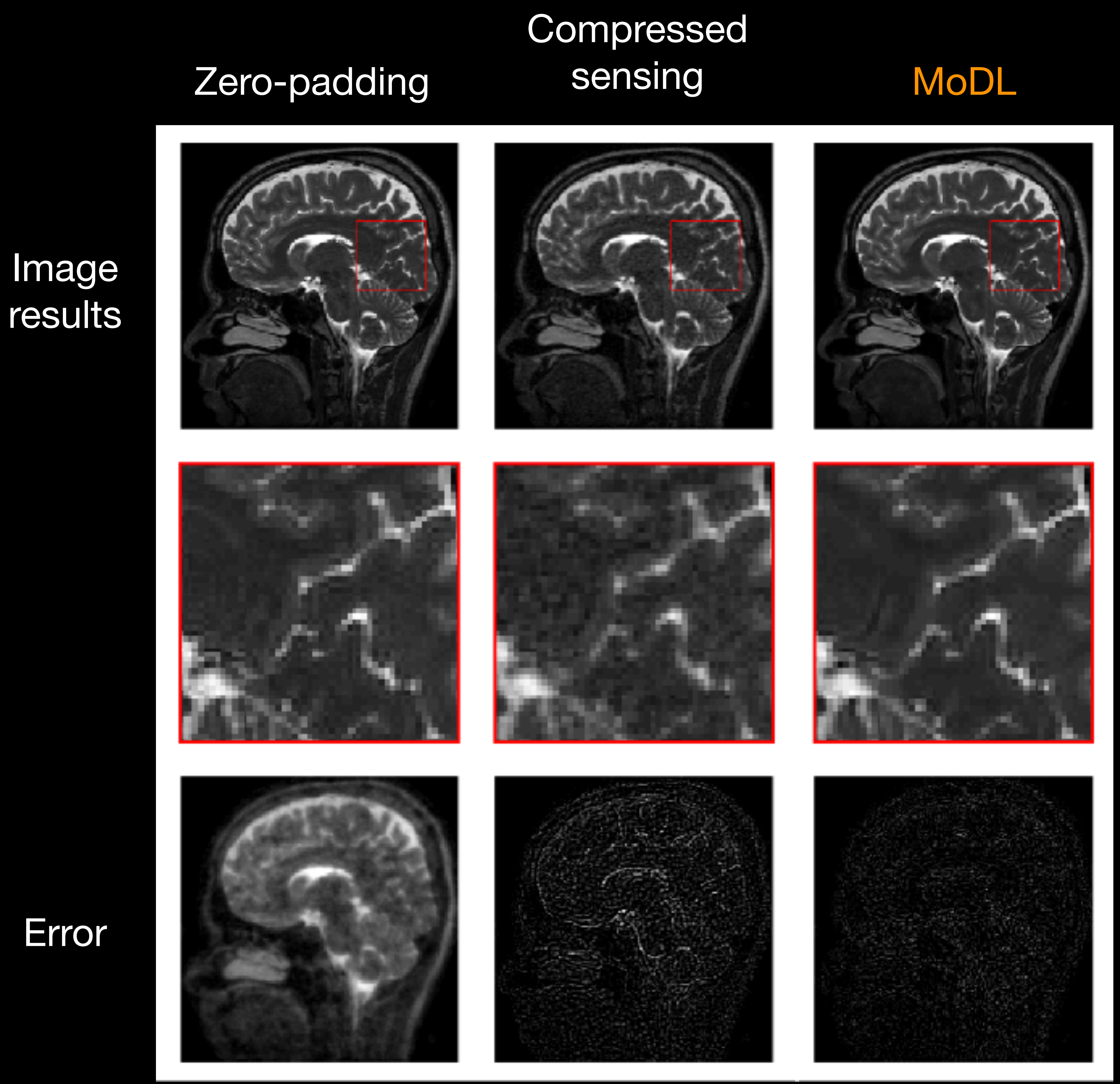
(1) MoDL

Overall MoDL architecture



k-space sampling pattern

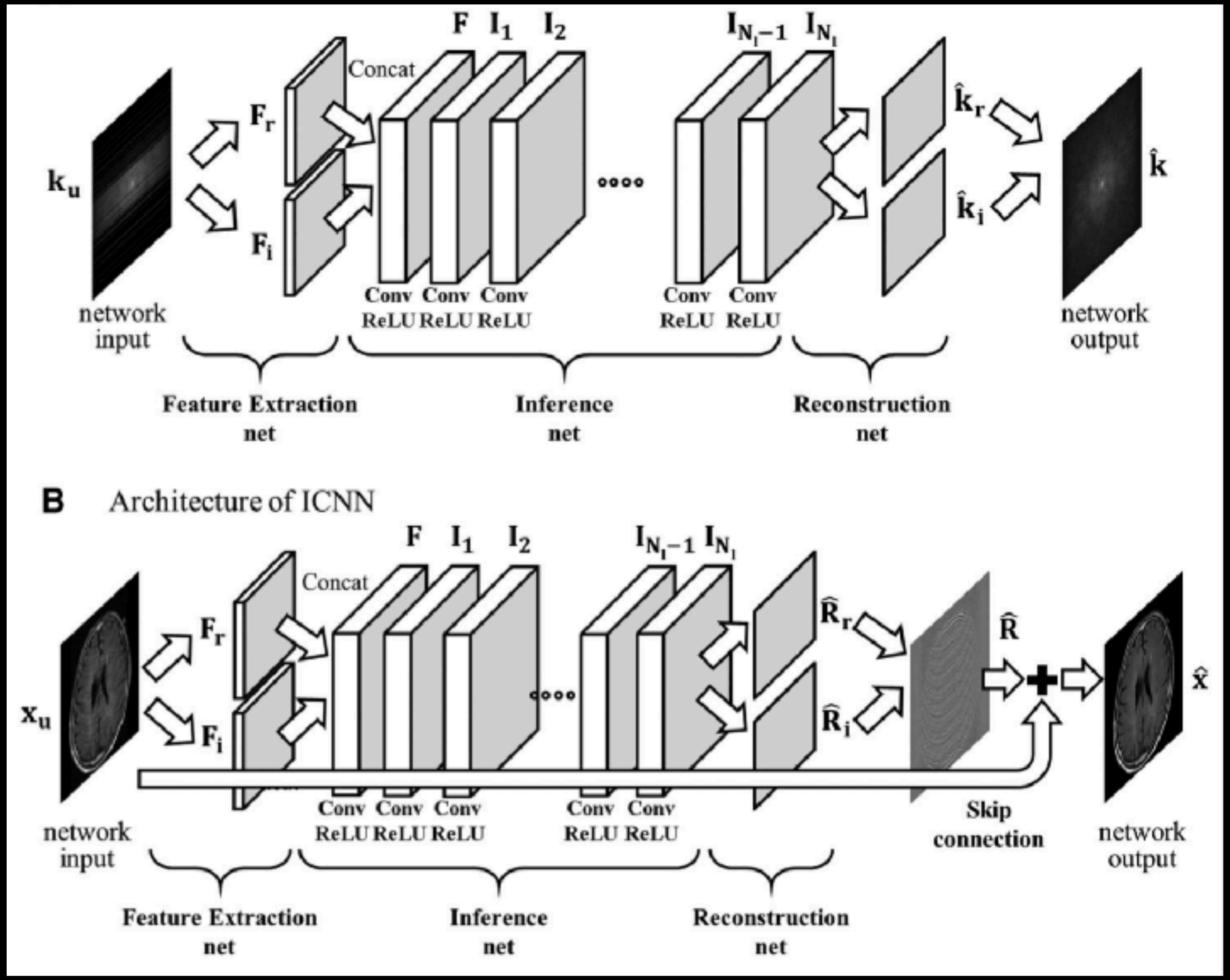


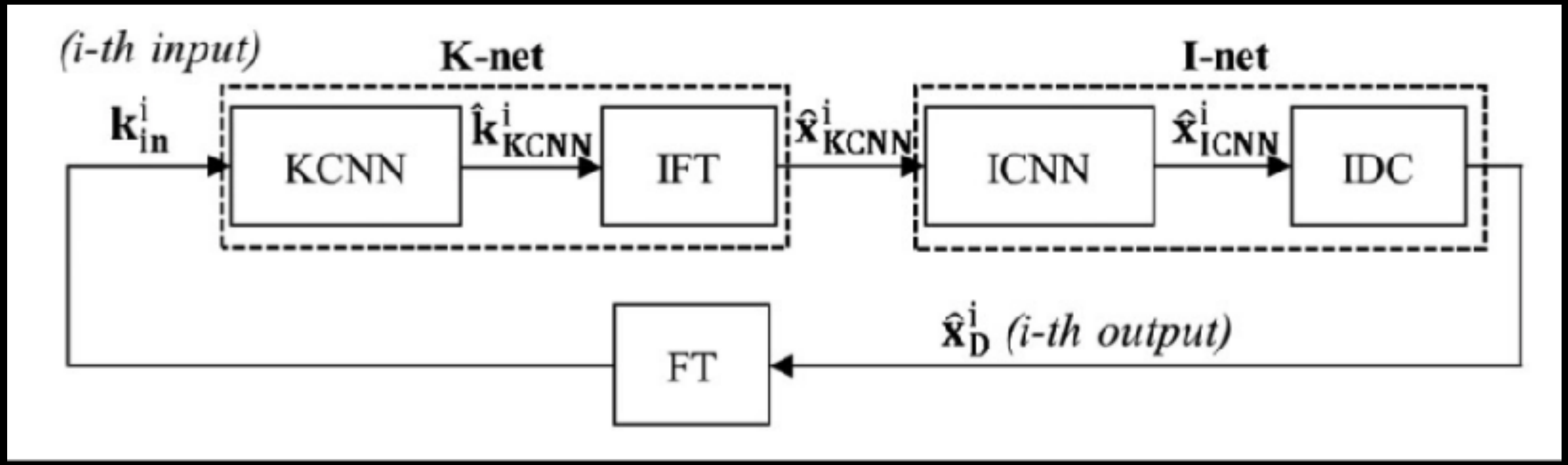


(Figures from: Aggarwal et al., IEEE TMI 2019)

(2) KIKI-net

- KIKI-net[1]: Use cross-domain ConvNets for image reconstruction
  - One sub-network for k-space completion
  - One sub-network for image restoration

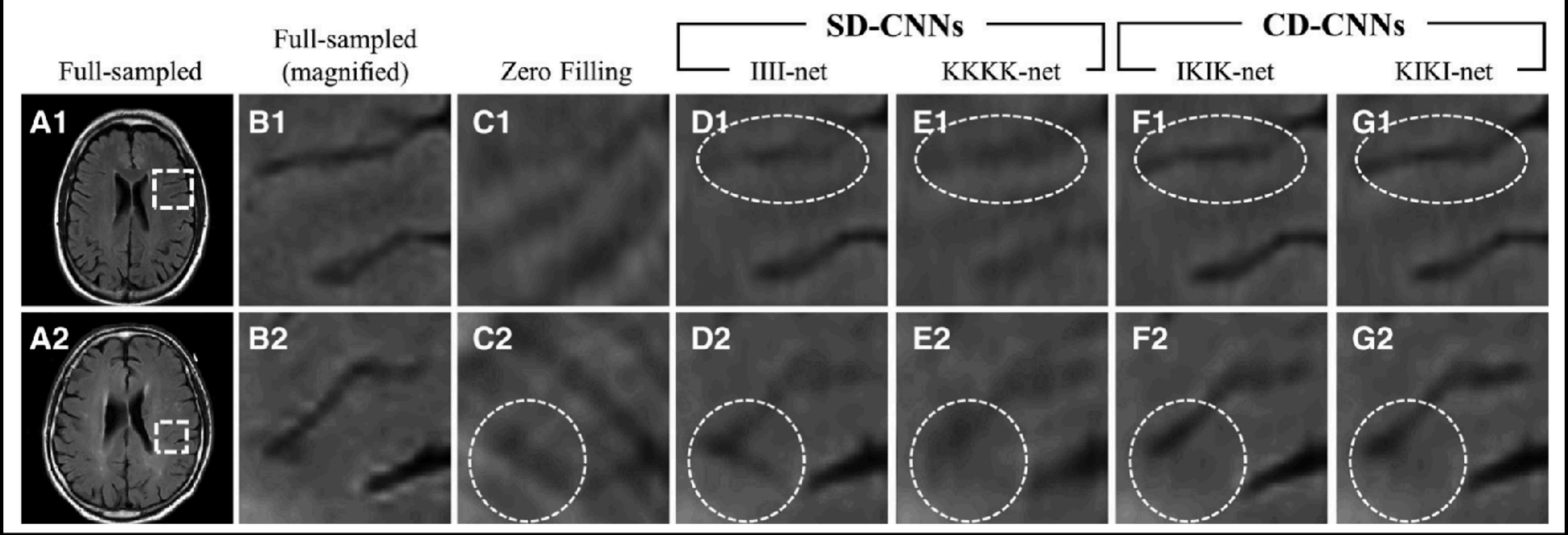




[1] Eo et al., MRM, 2018

(2) KIKI-net

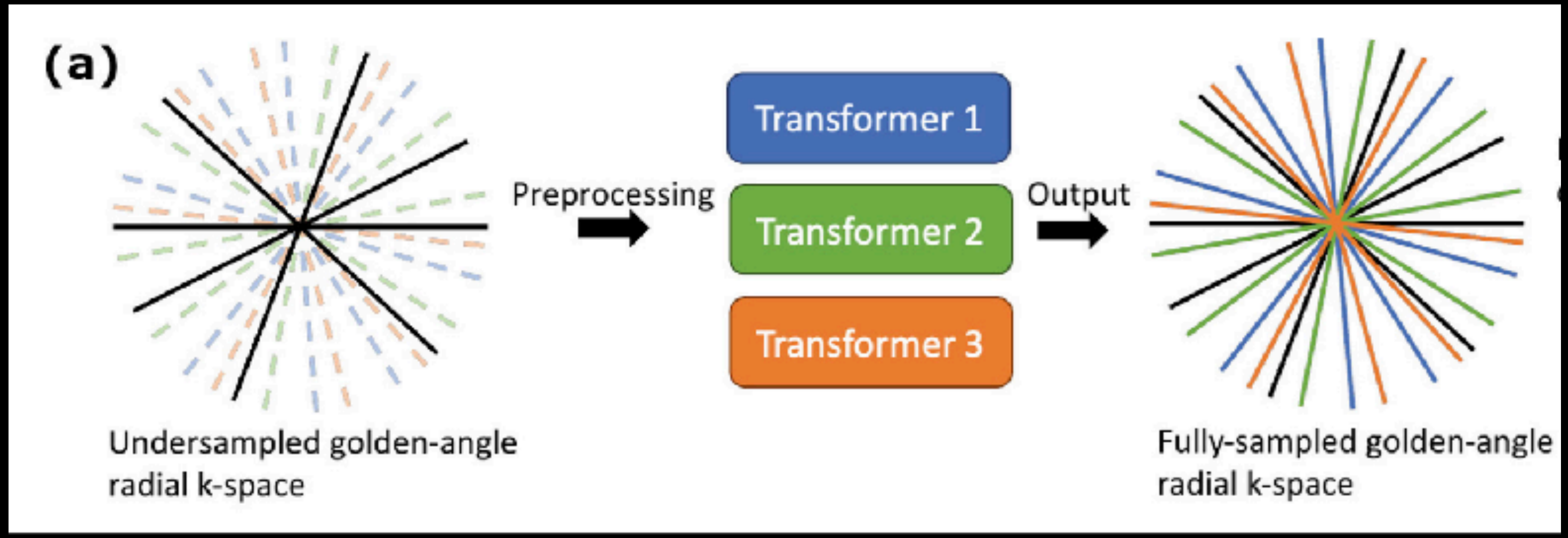
Results from single-domain CNN vs. cross-domain CNN
(undersampled factor R=4)

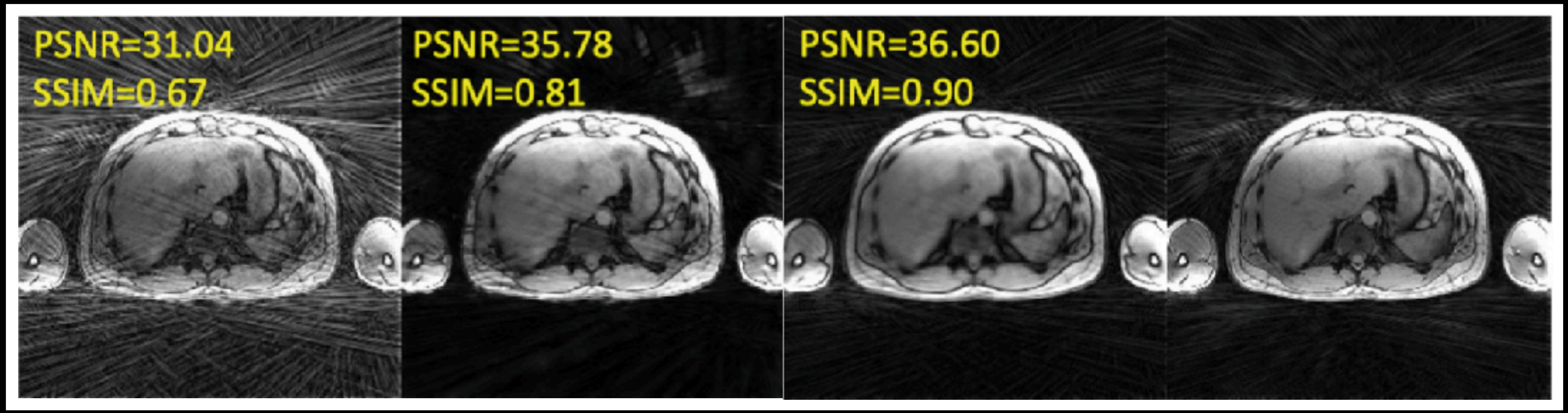


(Figure from: Eo et al., MRM 2018)

(3) PKT

- PKT[1] (projection-based k-space transformer):
  - Use a transformer network with self-attention mechanism to predict missing k-space spokes in radial MRI

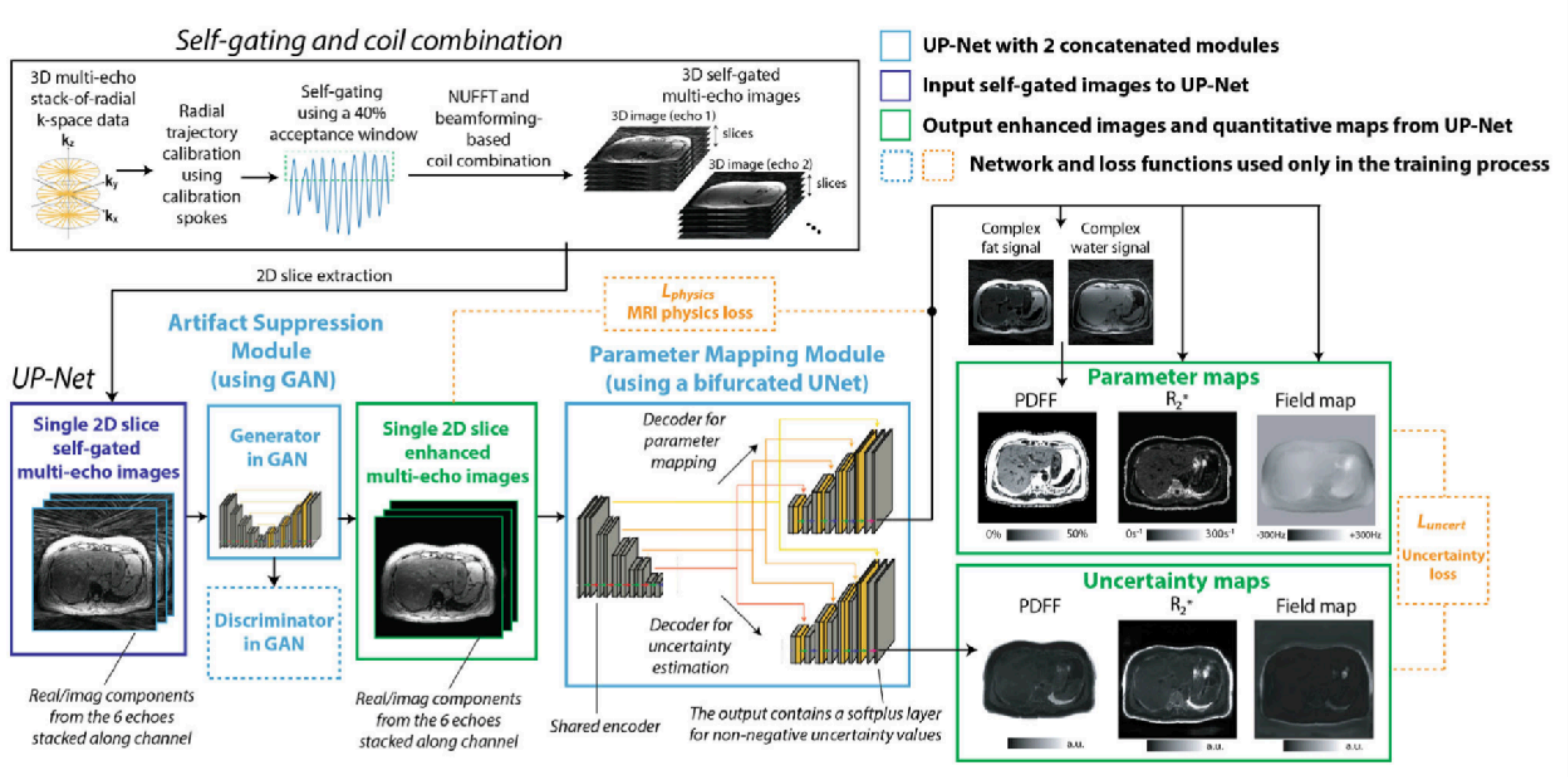




[1] Chang et al., MICCAI, 2022

(4) UP-Net

- UP-Net (Uncertainty-aware Physics-driven deep learning network)
  - Uncertainty information incorporated into deep learning-based artifact suppression and parameter mapping

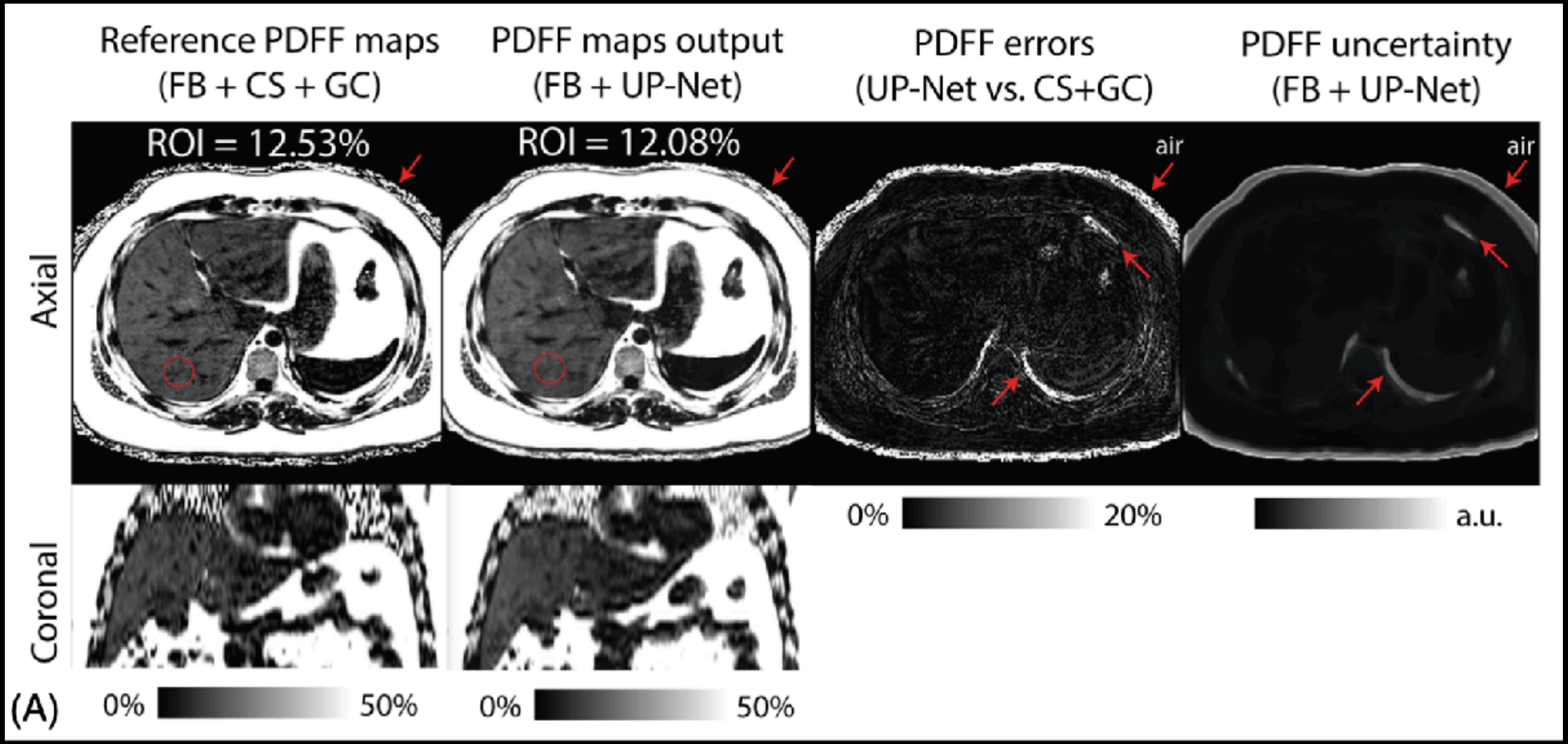


$$L_{\mathrm{uncert}} = \frac{\|\hat{p} - p\|_1}{\hat{u}} + \log(\hat{u}).$$

[1] Shih et al., MRM, 2023

(4) UP-Net

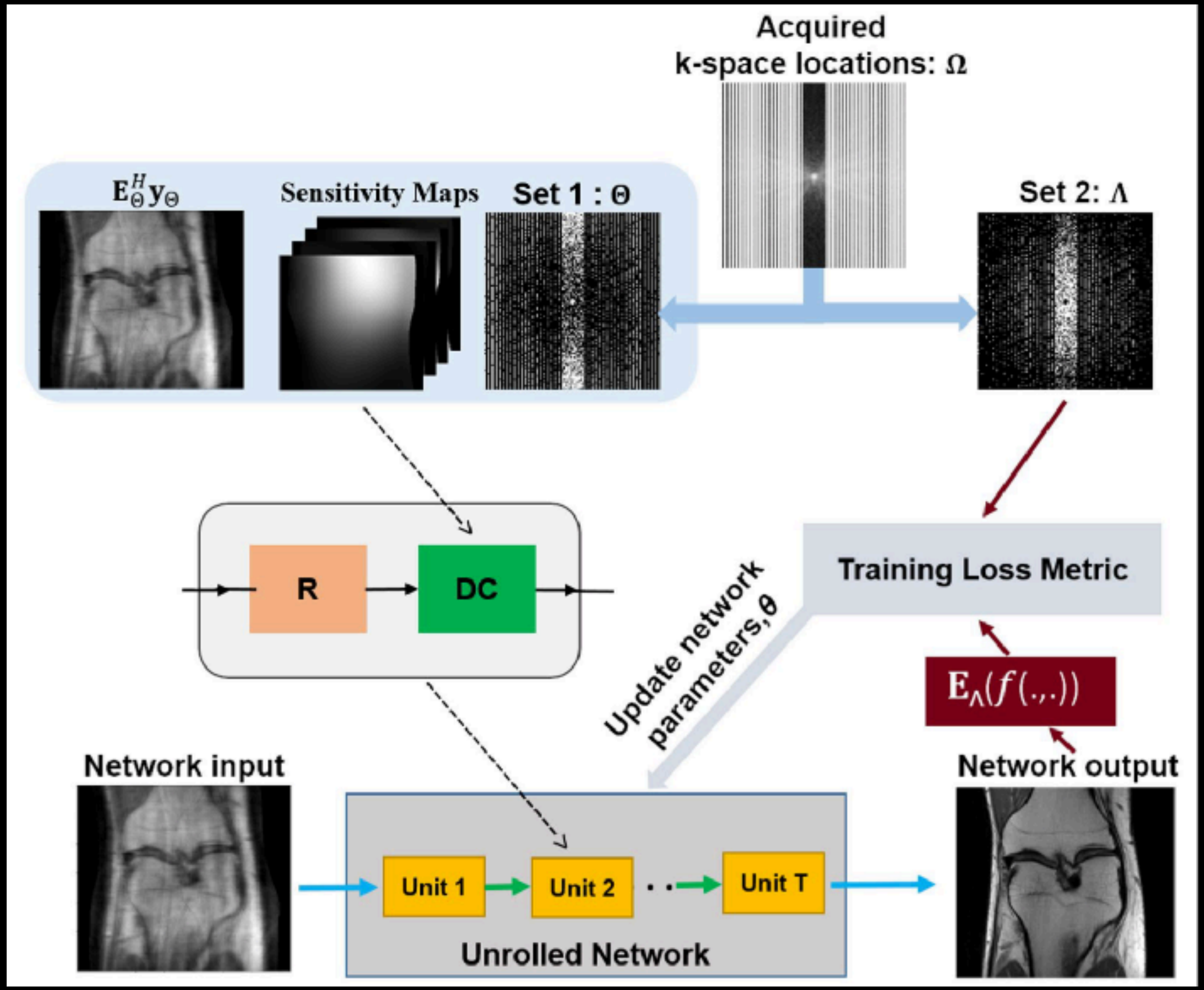
- Additional uncertainty map provided by the deep learning network can be used to estimate errors in the deep learning results



(Figure from: Shih et al., MRM 2023)

(5) Self-supervised physics-guided reconstruction

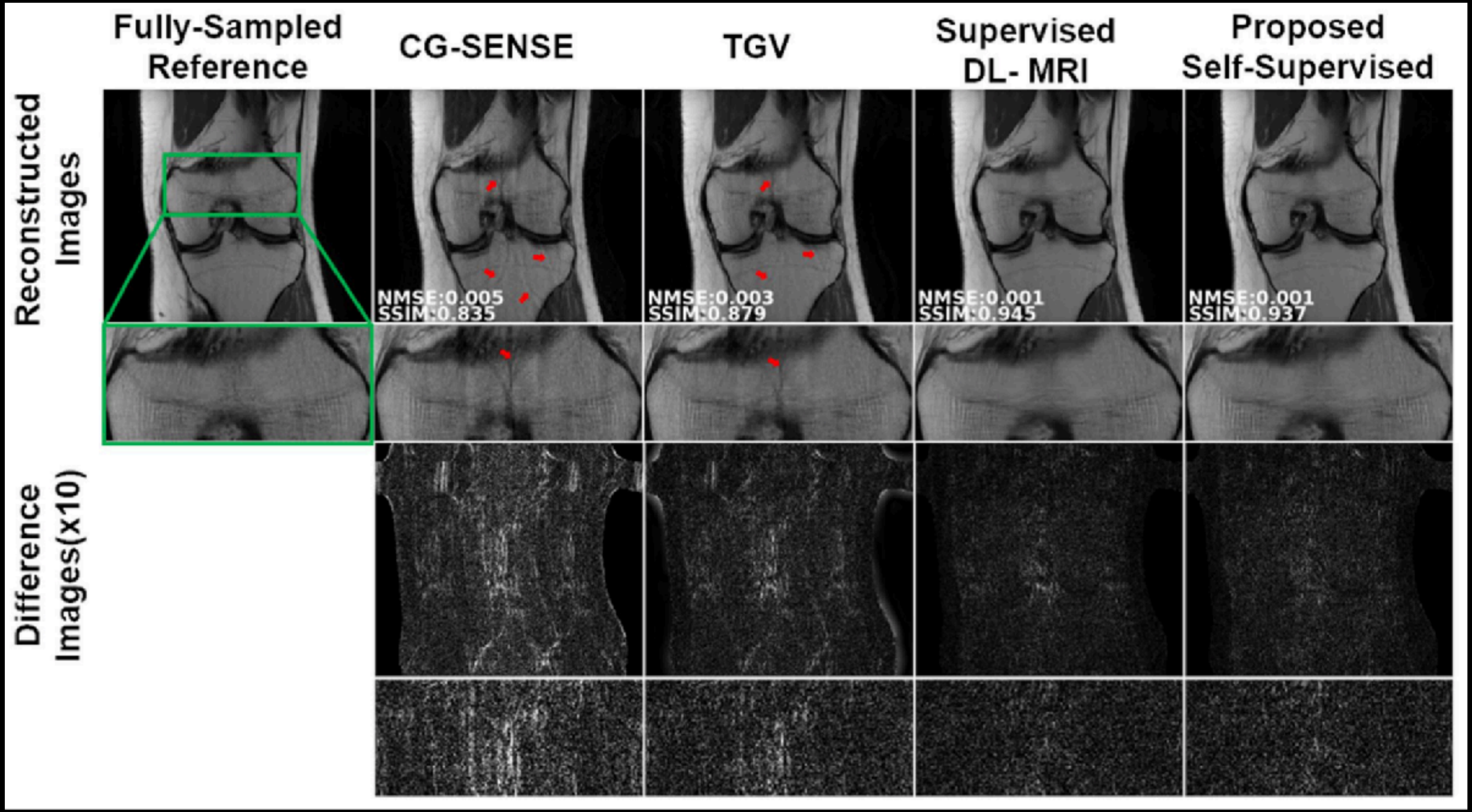
- Self-supervised physics-guided reconstruction[1]
  - Deep learning reconstruction without fully-sampled reference dataset
  - Acquired k-space was split into 2 disjoint sets for self-supervision during training.



[1] Yaman et al., MRM, 2020

(5) Self-supervised physics-guided reconstruction

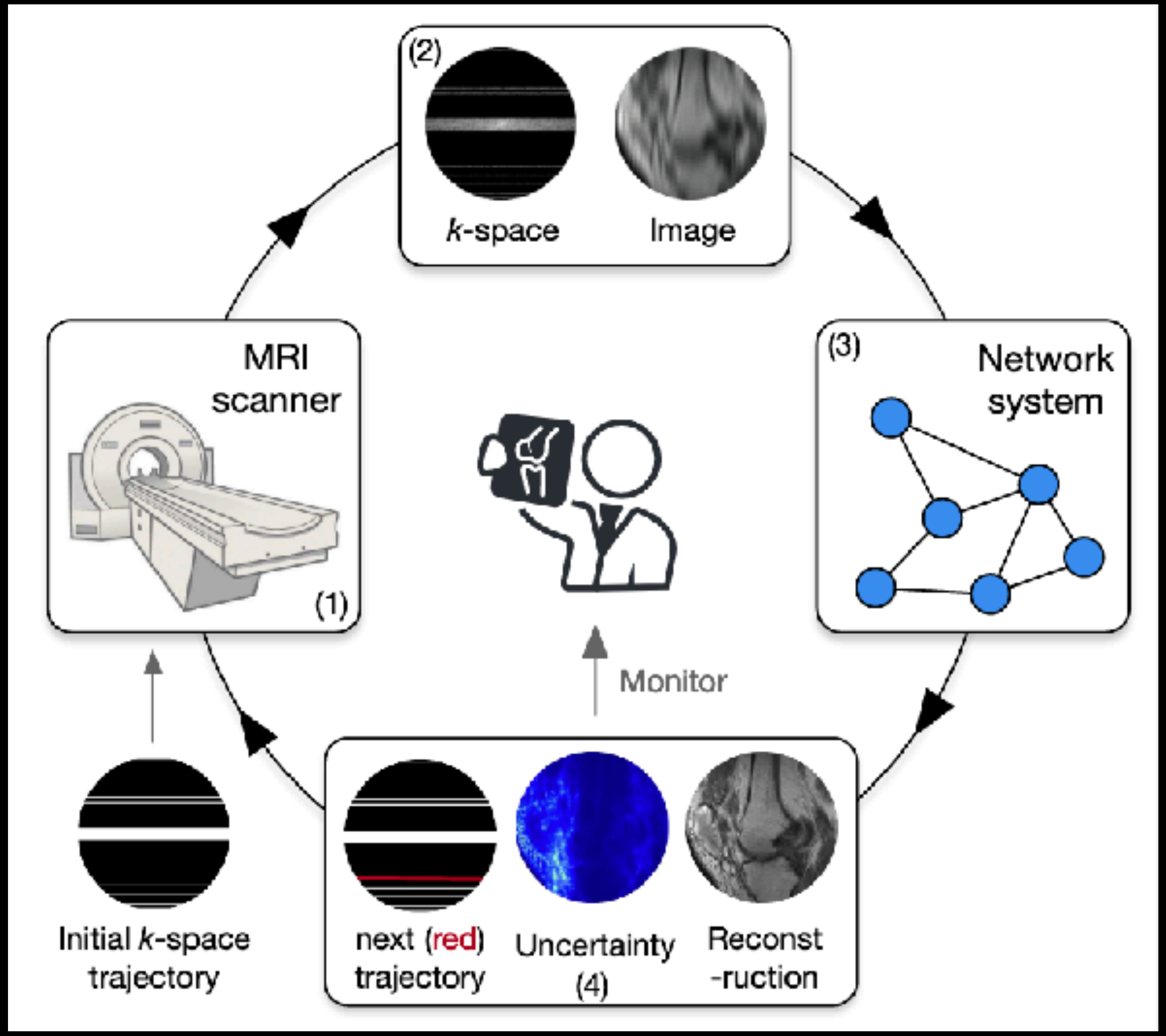
- Image from self-supervised learning show similar performance compared to the supervised method.



(Figure from: Yaman et al., MRM 2020)

(6) Active MRI acquisition

- Active MRI acquisition
  - Develop an *evaluator network* to rate the reconstruction uncertainty and the quality gain after each k-space line measurement
  - It is trained jointly with a *reconstruction network*.



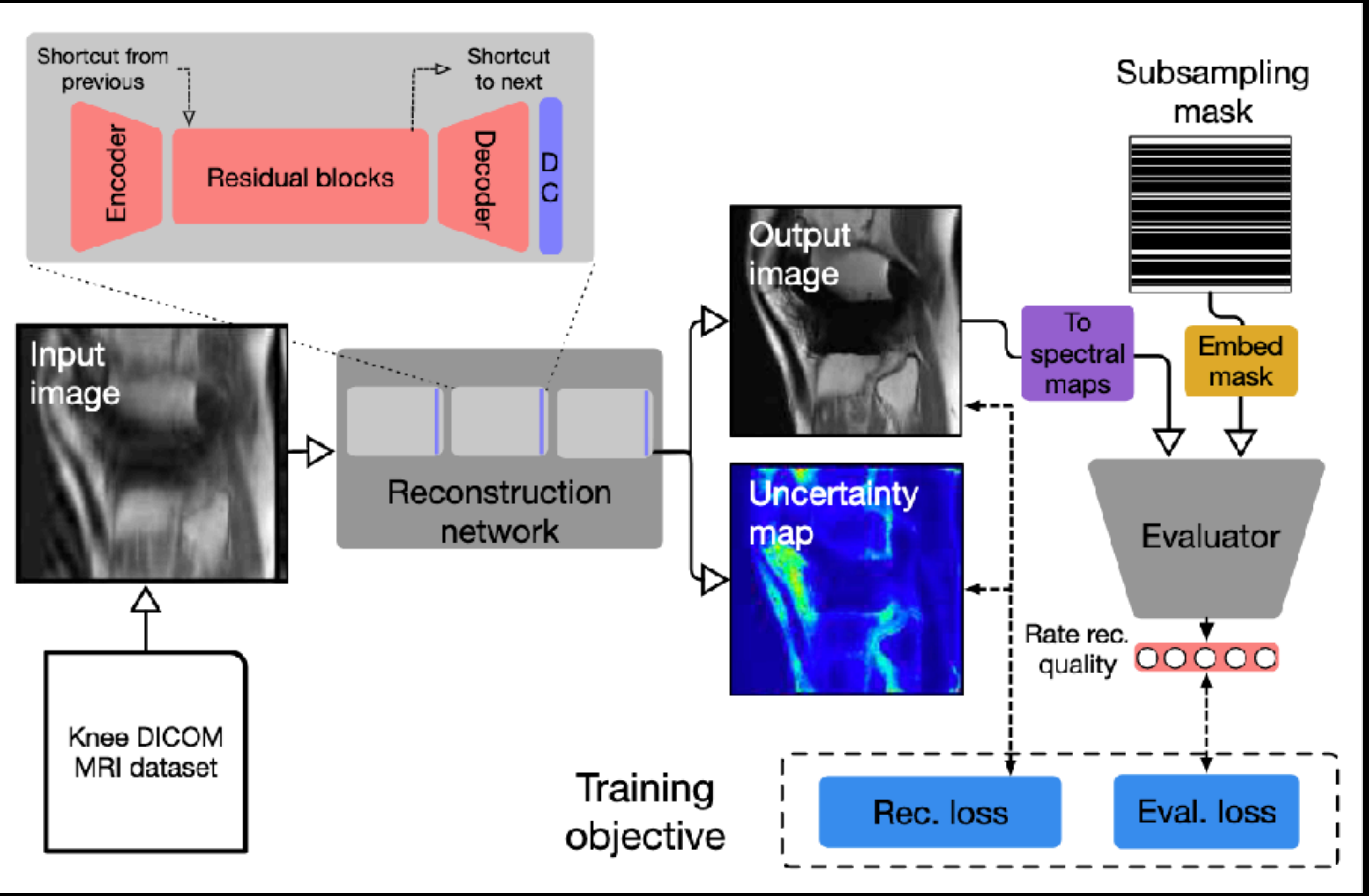
[1] Zhang et al., CVPR, 2019

(6) Active MRI acquisition

- The same undersampling pattern may not be suitable for all the cases.
- Deep learning-based uncertainty map was used to decide if sufficient data is acquired for faithful reconstruction.

For uncertainty estimation:

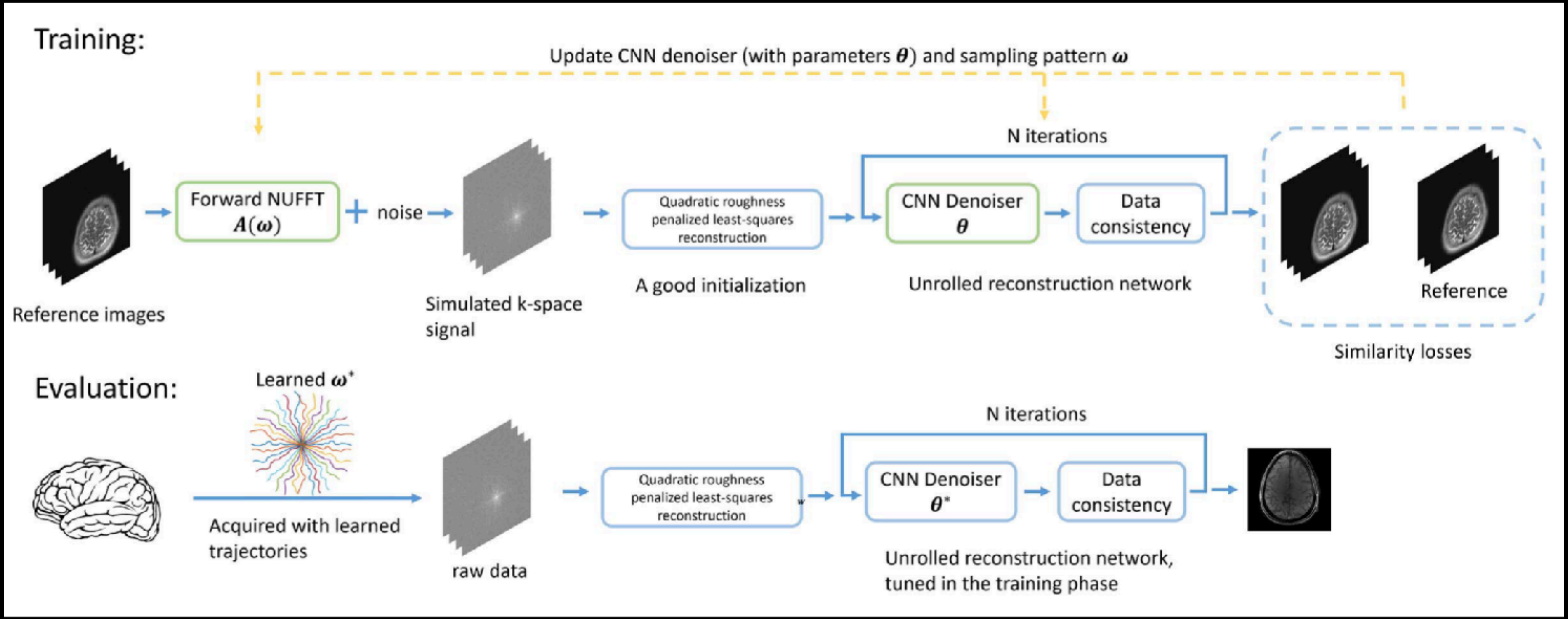
$$\mathcal{L}_R(\hat{\mathbf{x}}, \mathbf{r}, \mathbf{x}) = \frac{1}{N^2} \sum_{i=1}^{N^2} \frac{|\mathbf{r}_i - \mathbf{x}_i|^2}{2u(\hat{\mathbf{x}})_i} + \frac{1}{2} \log(2\pi u(\hat{\mathbf{x}})_i)$$



(Figures from: Zhang et al., CVPR 2019)

(7) Joint reconstruction and trajectory optimization

- BJORK[1] (B-spline parameterized Joint Optimizations of Reconstruction and K-space trajectory)
  - Use deep learning to find a suitable k-space trajectory for undersampled MRI reconstruction



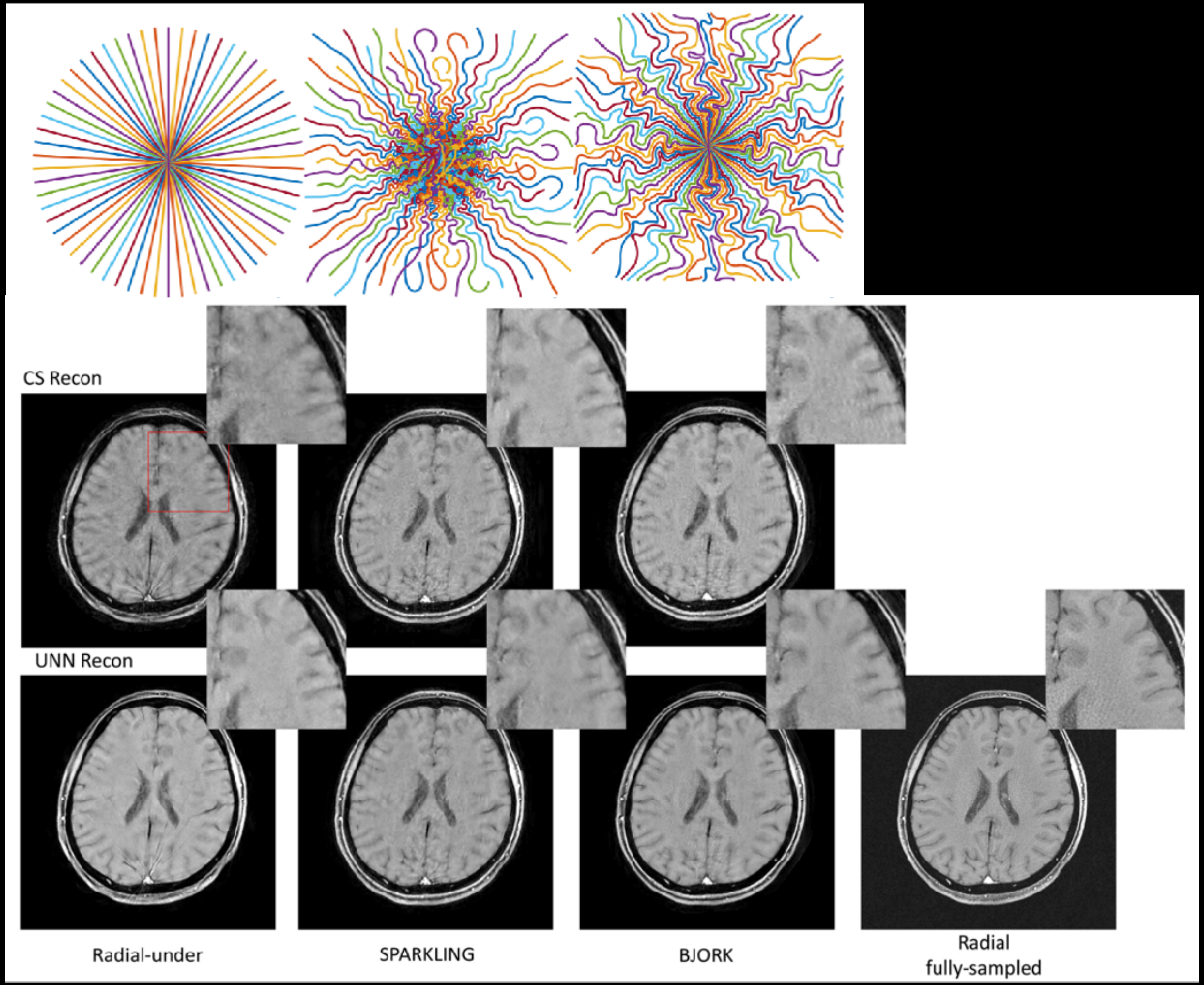
[1] Wang et al., IEEE TMI, 2022

(7) Joint reconstruction and trajectory optimization

Radial MRI and learned trajectory

CS reconstruction

DL reconstruction



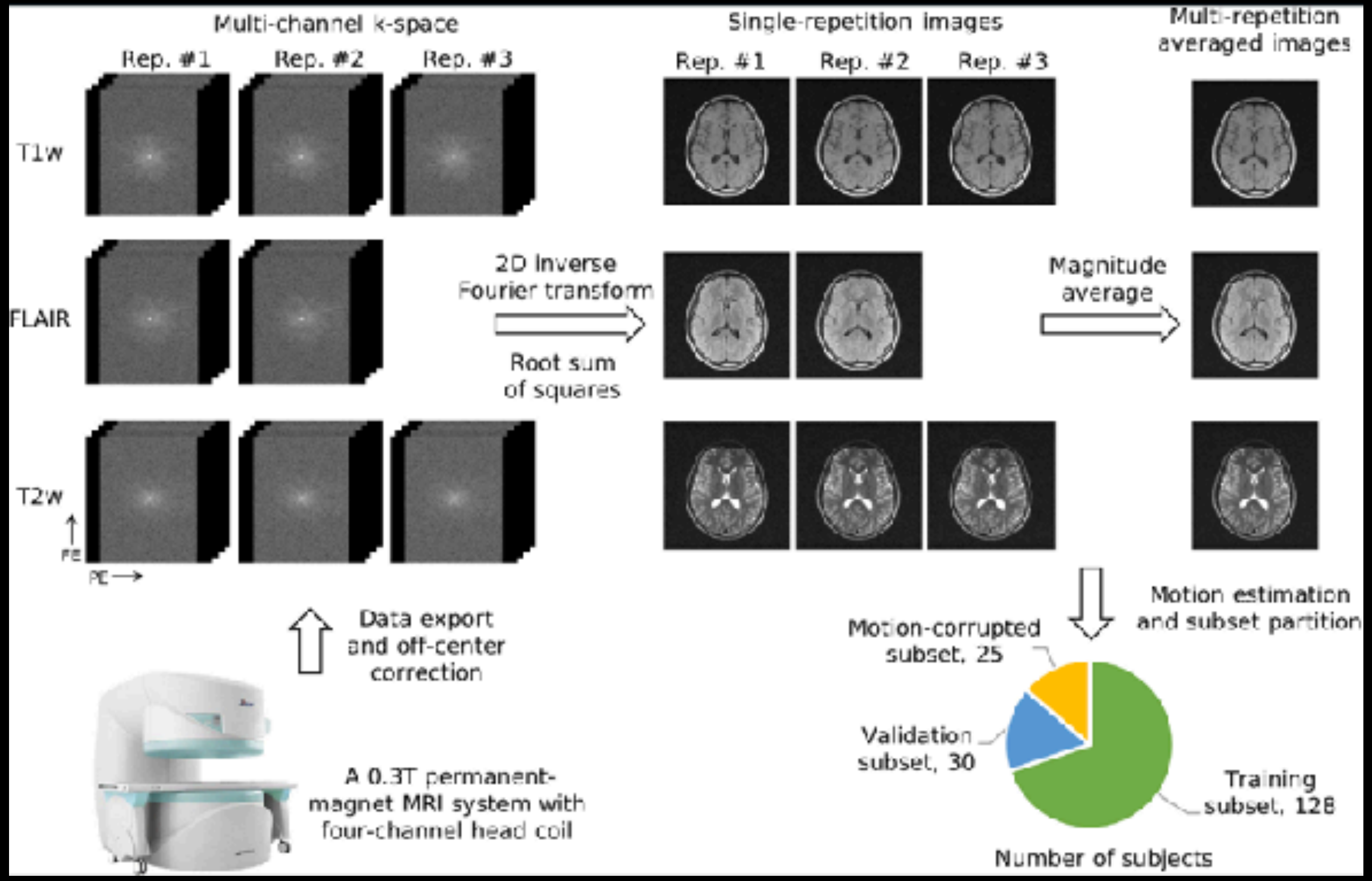
(Figures from: Wang et al., IEEE TMI 2022)

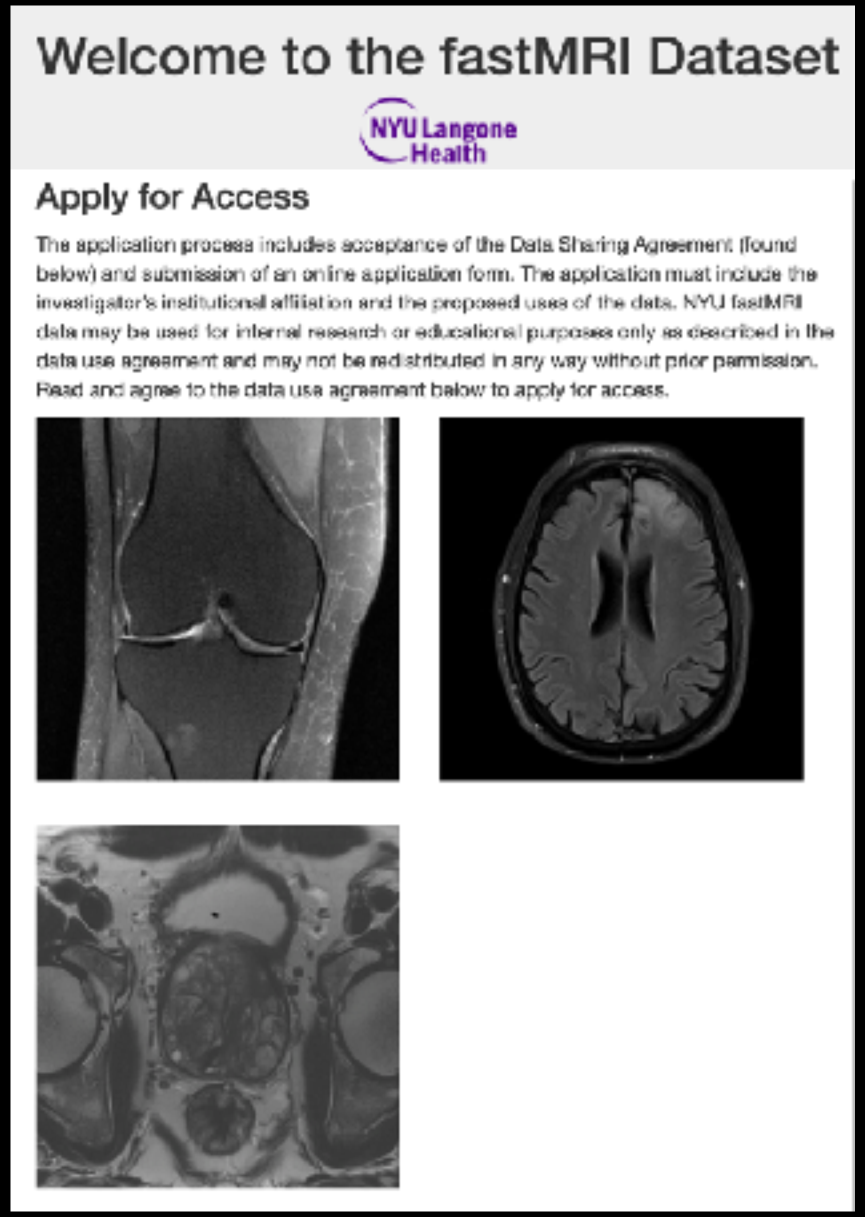
Deep learning MRI reconstruction

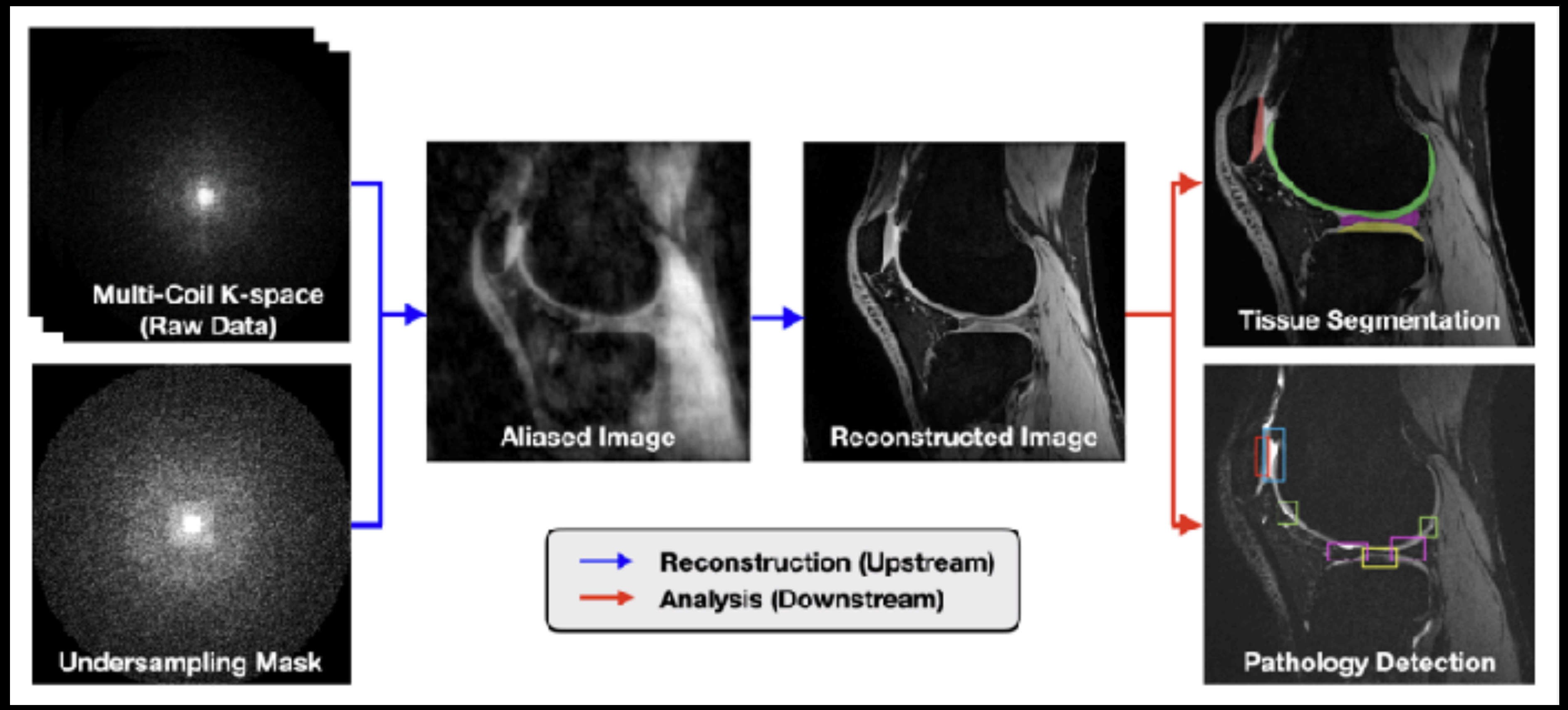
- Deep learning neural networks are tailored for specific applications to solve specific problems.
- Many of the methods involve k-space data at some point. It can softly constrain the reconstruction results to be consistent with the acquired undersampled k-space data.
- Some applications requires multi-tasking (e.g., reconstruction + uncertainty estimation)

Publicly available datasets for MRI reconstruction

- Public datasets with k-space data available
  - fastMRI (https://github.com/facebookresearch/fastMRI)
    - Knee, brain and prostate MRI
  - SKM-TEA (https://github.com/StanfordMIMI/skm-tea)
    - Quantitative knee MRI with tissue segmentation
  - M4Raw (https://github.com/mylyu/M4Raw)
    - Multi-contrast multi-repetition 0.3T brain MRI
  - … and more



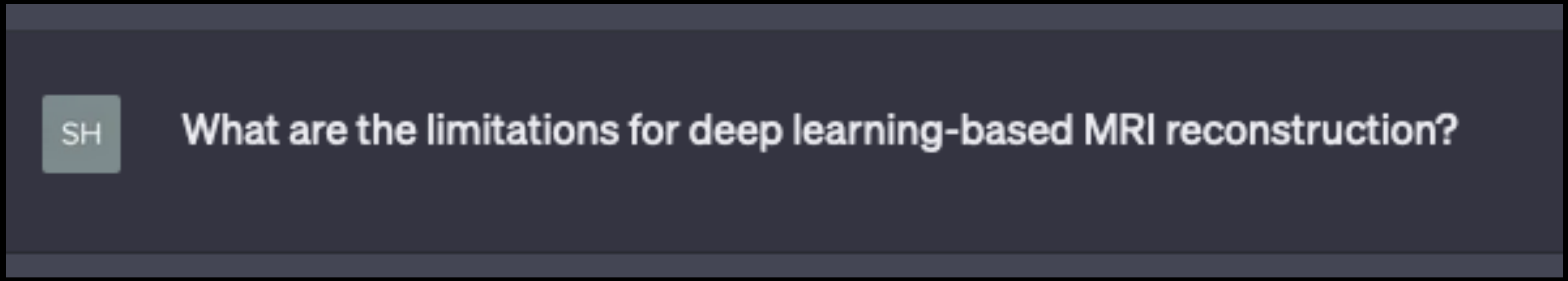




(Figures from: https://fastmri.med.nyu.edu/, Desai et al., arxiv 2022, https://github.com/mylyu/M4Raw)

Discussion

- All major MRI vendors are working on deep learning-based MRI reconstruction
- There are many opportunities, but there are also many open questions.
- What are the limitations for deep learning-based MRI reconstruction?
  - Let’s ask ChatGPT…



Discussion

- Limitations of deep learning-based MRI reconstruction
  - Insufficient training data
    - Even though there are public large datasets, obtaining diverse and representative dataset is still challenging.
  - Lack of interpretability / “Failure mode” not clear
    - The black-box nature of deep learning can be problematic for clinical acceptance and trust.
    - Uncertainty quantification or theory to explain deep learning are being investigated
  - Generalization to different acquisition parameters
    - Potential solution would be including large datasets with all different acquisition parameters or including sequence parameters as inputs
  - Computational complexity
    - The hardware keeps advancing and it can still be expensive

A few *personal* suggestions…

- Focus on the problem you want to solve (*to improve image quality? to allow for higher undersampling factors? to train without ground truth images?…*).

- Have a good understanding on the deep learning tools you have. Choose or develop methods or architectures that can solve the problem.

- Understand your data and be aware of the MRI signal model and acquisition process. There can be constraints or there can be some prior information to utilize.

- Don’t get lost in numbers! Don’t forget the clinical problem.

Thanks!

- Next time:
  - Managing Motion in MRI by Dr. Wu

Shu-Fu Shih

sshih@mednet.ucla.edu