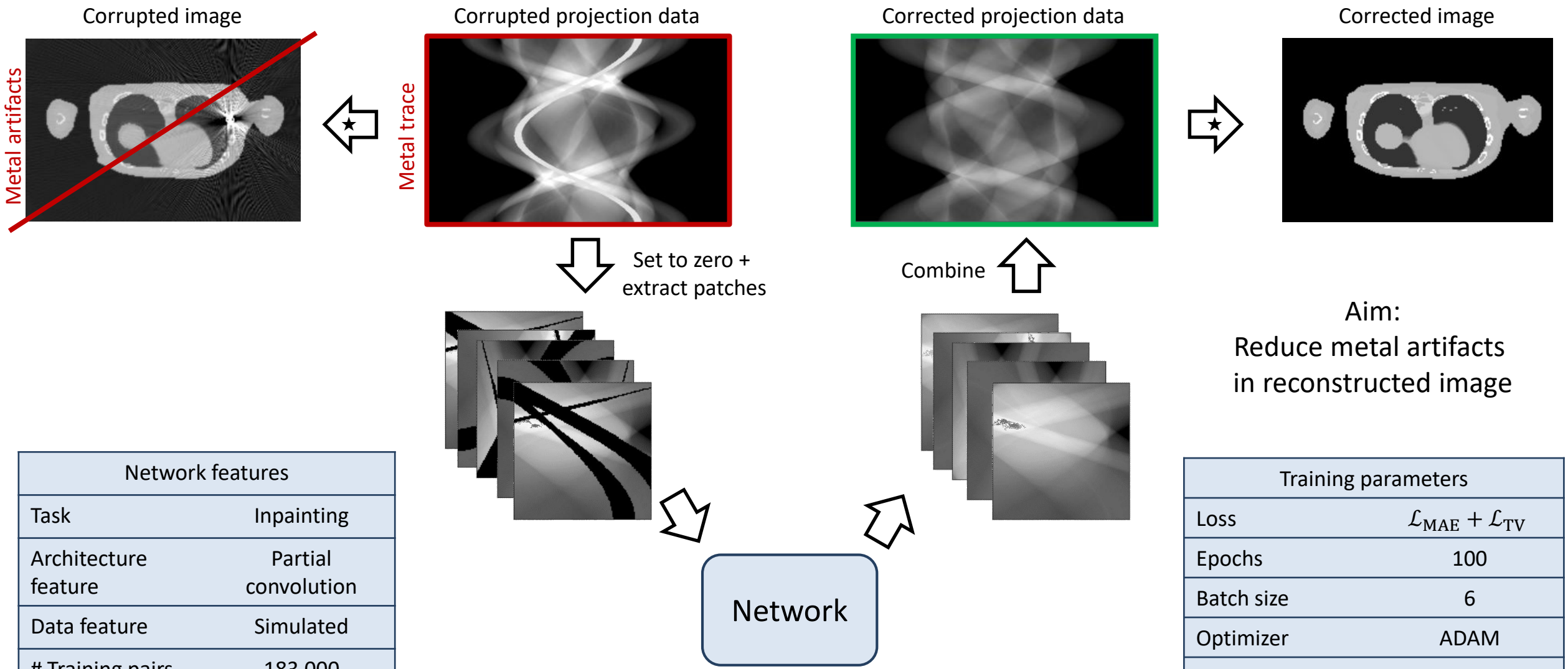
Laura Hellwege[1], Nele Blum[1], Thorsten Buzug[1,2] and Maik Stille[1,2]

Motivation

★ : Reconstruction



Aim:
Reduce metal artifacts
in reconstructed image

Network features	
Task	Inpainting
Architecture feature	Partial convolution
Data feature	Simulated
# Training pairs	183 000

Training parameters	
Loss	$\mathcal{L}_{\mathrm{MAE}} + \mathcal{L}_{\mathrm{TV}}$
Epochs	100
Batch size	6
Optimizer	ADAM
Learning rate	2e-5

[1] Institute of Medical Engineering, University of Lübeck, Germany [2] Fraunhofer Research Institution for Invidualized and Cell-Based Medical Engineering, Lübeck, Germany

Laura Hellwege[1], Nele Blum[1], Thorsten Buzug[1,2] and Maik Stille[1,2]

Institut für Medizintechnik

Partial Convolution Network

Convolution Layer

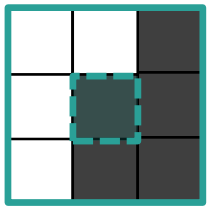
$$x'_{(i,j)} = \begin{cases} \mathbf{W}^T(\mathbf{X}_{(i,j)} \odot \mathbf{M}_{(i,j)})r_{(i,j)}, & \text{if } \|\mathbf{M}_{(i,j)}\|_1 > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$r_{(i,j)} = \frac{\|\mathbf{1}_{(i,j)}\|_1}{\|\mathbf{M}_{(i,j)}\|_1}$$

$$m'_{(i,j)} = \begin{cases} 1, & \text{if } \|\mathbf{M}_{(i,j)}\|_1 > 0 \\ 0, & \text{otherwise} \end{cases}$$

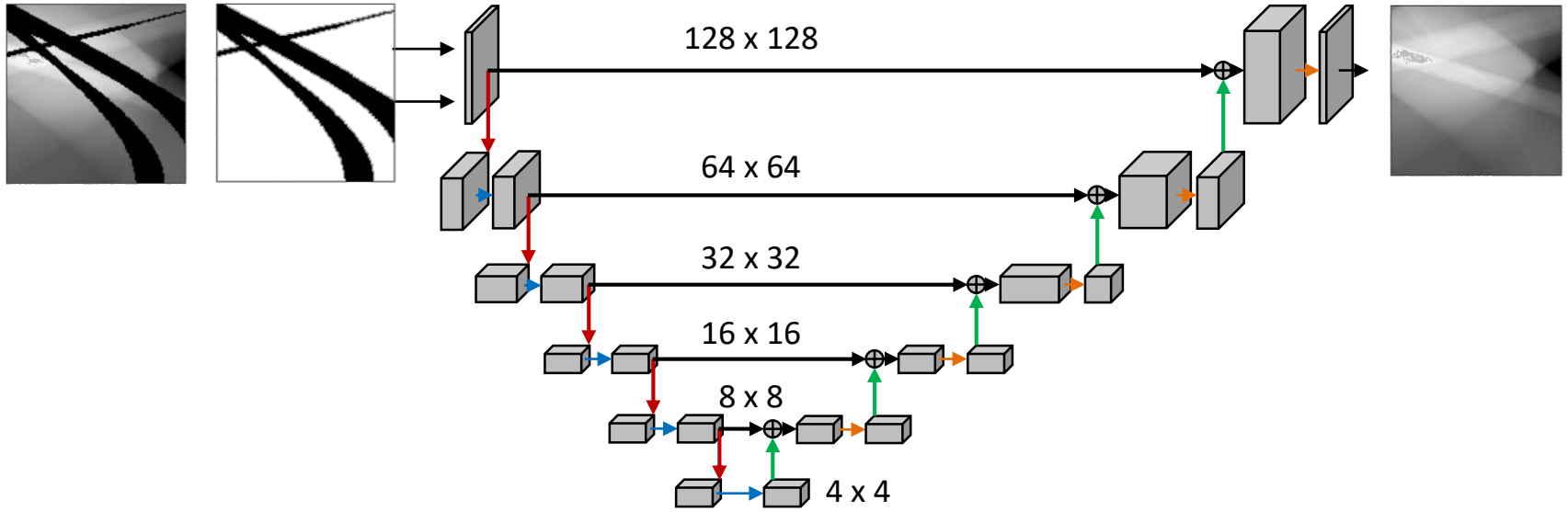
$\mathbf{W}$	Kernel
$\mathbf{X}_{(i,j)}$	Feature map under kernel
$\mathbf{M}_{(i,j)}$	Mask under kernel
$\odot$	Pointwise multiplication

Example:



$$\|\mathbf{M}_{(i,j)}\| = 4, r = \frac{9}{4}$$

Architecture



PC (Stride: 2) + ReLU

Batch Normalization

Upsampling

PC + LeakyReLU

Concatenation

Liu et al. Image Inpainting for Irregular Holes Using Partial Convolutions. In *The European Conference on Computer Vision* (2018).

Laura Hellwege[1], Nele Blum[1], Thorsten Buzug[1,2] and Maik Stille[1,2]

Institut für Medizintechnik

Results

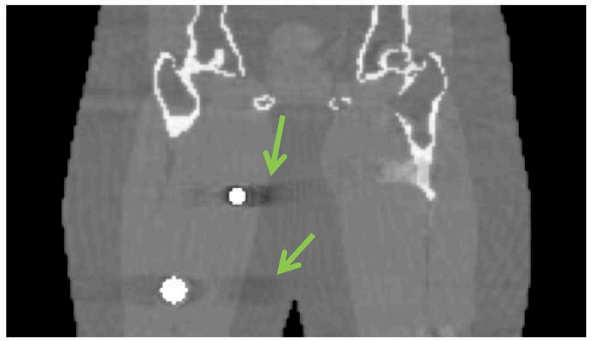
Method	MSE	MAE	SSIM
BLI	4.4E-3 / 4.61	3.8E-2 / 1.23	0.993 / 0.997
CNN	1.1E-2 / 137.13	6.4E-2 / 6.13	0.987 / 0.986
PCNN	9.8E-3 / 81.50	6.6E-2 / 5.22	0.987 / 0.990

BLI Bilinear Interpolation
CNN Convolutional Neural Network
PCNN Partial Convolution Neural Network

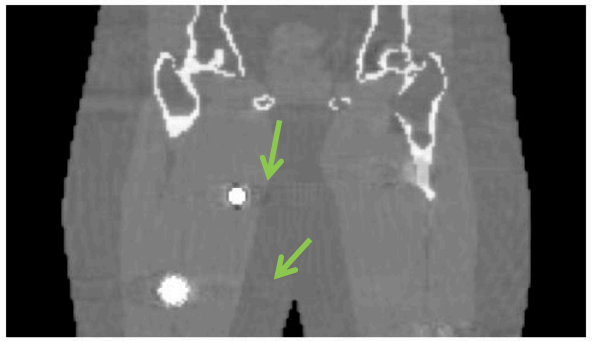
Projection domain

Image domain

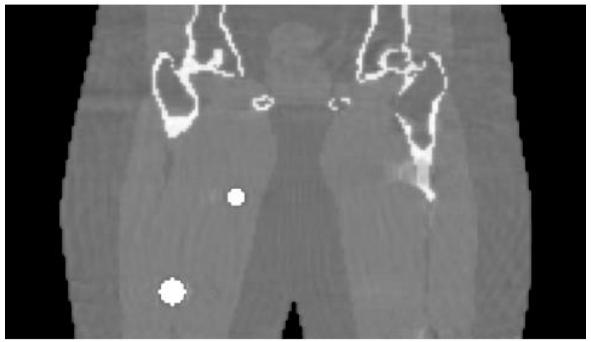
Level: 300 HU
Window: 500 HU



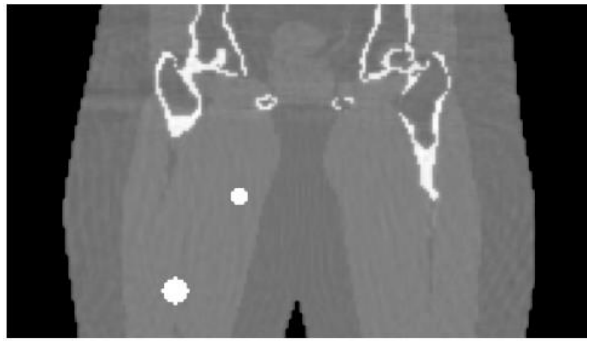
CNN



PCNN



BLI



Ground-truth

- CNN and PCNN perform similarly in the projection domain
- PCNN outperforms CNN in image domain
- Shown results can be improved by change of architecture and training parameters