## Online supplementary material

### Online supplement A: Augmentations

Between brackets the probability of augmentation per BL-FU NCCT slice pair is given. Implementations are available on: github.com/henkvanvoorst92/FU2BL-GAN

Orientation: horizontal flipping (p=0.5), Random rotation between -20 and 20 degrees (p=0.5).

Intensity (p=0.2 for one of the following): Adding of HU at random between -20 and 20, multiplying all HU with a factor between 0.8 and 1.2, gamma scaling within a gamma range of 0.9;1.1 and a gain of 1 (implementation skimage.exposure.adjust_gamma).

Noise: Adding of an unsharp noise mask multiplied with a random factor between 0.5 and 1.2 (p=0.2), gaussian filtering (p=0.2), adding of beam hardening noise multiplied with a factor between -0.8;-0.2 and 0.2;0.8 (p=0.2). Beam hardening noise was generated by using the filtered back propagation algorithm with a ramp filter to generate a sinogram (skimage.transform.radon), the sinogram values were clipped to the mean and reconstructed back to an image (skimage.transform.iradon) that represents the beam hardening noise.

After the augmentations the data was clipped and normalized to a -1, 1 range.

### Online supplement B: Model architecture, training loss, validation DSC

The ResNet generator and patch discriminator architectures presented in the original pix2pix paper by Isola et al. were used.

**Generator model:** The generator model exists out of an encoder, 6 repeated resnet blocks, and a decoder. The first and last convolution layer of the network used a kernel of size 7x7 while all other layers used a kernel of size 3x3. After each convolution in the encoder and decoder network instance normalization was applied followed by a rectified linear unit (ReLu) activation layer, only for the output layer a Tanh activation layer was used without normalization. Reflection padding was used after each convolution, pad size=(kernel size -1)/2, to prevent information loss at the edges of the scans.

For the encoder the spatial resolution was halved three times by applying a strided convolution (stride=2x2) while the number of filters was increased with the following steps: 2, 64, 128, 256 filters. Subsequently, 6 resnet blocks were added with 256 filters per convolution operation, each block consisted of a 3x3 convolution, instance normalization, 0.5 dropout, ReLu activation, again a 3x3 convolution, and an instance normalization layer. The resnet blocks had skip connections where the output of the block was added to the input to generate the final output. The decoder networks used three transposed convolutions with stride 2x2 to increase the spatial resolution back to the original input size while reducing the number of filters with the following steps: 256, 128, 64, 1.

**Discriminator model**: Three convolution operations with kernel size 4x4 and stride 2x2 were used to halve the spatial resolution three times while increasing the number of filters from 64 to 128 to 256. Subsequently, two convolutions with kernel size 4x4 and stride 1x1 were increased the number of filters to 512 and decrease the filters to 1. After each convolution operation except the last one instance normalization and a leaky ReLU activation layer were added. The final layer used a sigmoid activation function to generate a probability output per patch used to compute the binary cross entropy loss. The average binary cross entropy loss over the patches was used as discriminator loss.

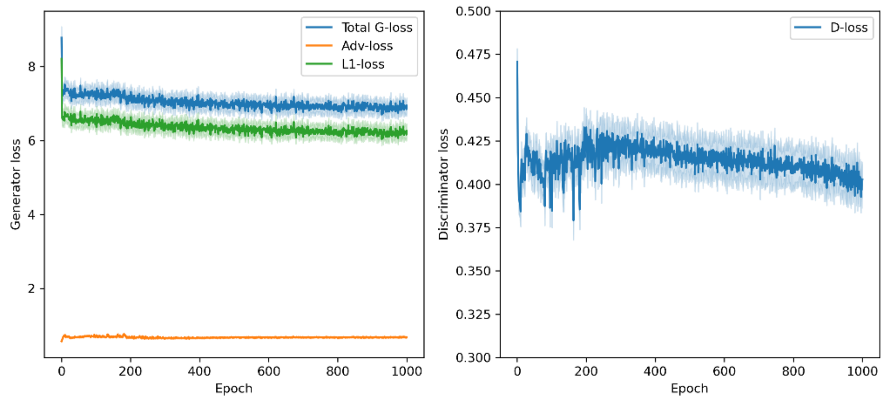
**Online supplement C: Training data characteristics**

| | Baseline | <8H | 24H | 1W |
|---|---|---|---|---|
| Number of inclusions | 820 | 115 | 681 | 415 |
| Slice thickness in mm <1/1-3/3-5/>5 count(%) | 466(57)/230(28)/122(15)/2(0.2) | 86(75)/21(18)/7(6)/1(0.9) | 448(66)/181(27)/46(7)/6(1) | 247(60)/92(22)/67(16)/9(2) |
| Scan manufacturer Siemens/Philips/Toshiba/GE/Canon/unknown count(%) | 537(65)/159(19)/90(11)/33(4)/1(0.2)/0 | 88(77)/22(19)/0/5(4)/0/0 | 535(79)/113(17)/12(2)/21(3) | 287(69)/90(22)/23(6)/15(4) |
| Peak potential in kV 80/100/110/120/140/unknown count(%) | 5(1)/293(36)/16(2)/503(61)/3(0.4)/0 | 51(44)/28(24)/2(2)/27(23)/0/7(6) | 95(14)/186(27)/18(3)/255(37)/25(4)/102(15) | 8(2)/144(35)/13(3)/247(60)/1(0.2)/2(0.5) |
| Exposure in mAs mean(std) | 319 (129) | 353 (129) | 275 (129) | 309 (129) |
| CT dose index volume in mGy mean(std) | 42 (12) | 39 (12) | 38 (12) | 40 (12) |

**Online supplement D:**

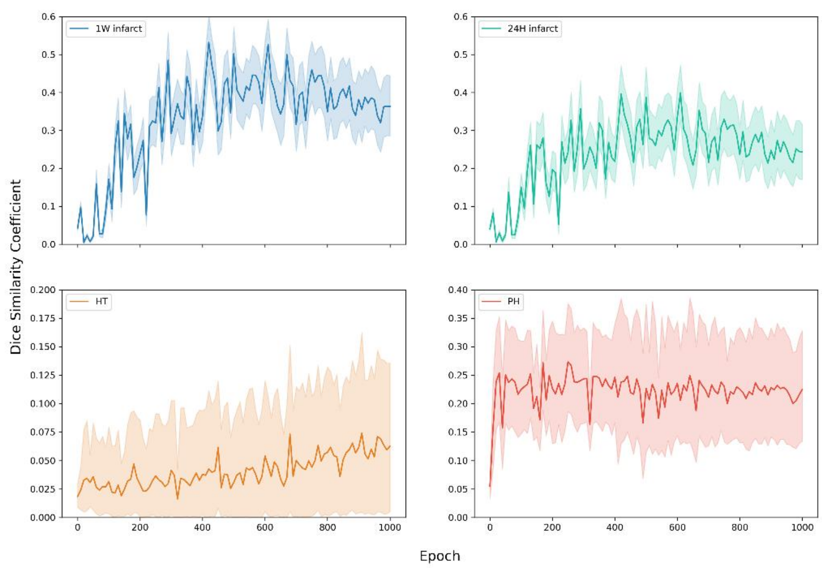
Training of the FU2BL-GAN took 67 hours for 1000 epochs.

*Figure D1. Training loss FU2BL-GAN: Total G-loss: total loss of generator existing out of adversarial loss (Adv-loss) and absolute voxel wise difference between generated and real baseline CT (L1-loss). D-loss: binary cross entropy loss of the discriminators prediction.*




The optimal epoch and difference map threshold was obtained by computing the Dice similarity coefficient (DSC) every 10$^{th}$ training epoch for thresholds lower than (infarct) and higher than (hemorrhage) values in the range of -0.2 to +0.3 with steps of 0.01 (equivalent to 0.5 HU). No early stopping criterion was used due to the linearly decreasing learning schedule (after epoch 500) and the unstable nature of the GAN optimization.

*Figure D2. Validation dice similarity coefficient every 10$^{th}$ epoch for optimal thresholds*




| Dataset | Epoch | Threshold |
|---|---|---|
| 24H infarct lesion | 610 | <-0.10 (5 HU) |
| 1W infarct lesion | 420 | <-0.10 (5 HU) |
| HT lesion | 880 | >0.11 (5.5 HU) |
| PrimHem lesion | 250 | >0.11 (5.5 HU) |

**Table B1: Optimal epoch and threshold of the FU2BL-GAN for segmentation.** The optimal epoch and threshold based on the validation set Dice similarity coefficient (DSC).

**Online supplement E: Supplementary results**

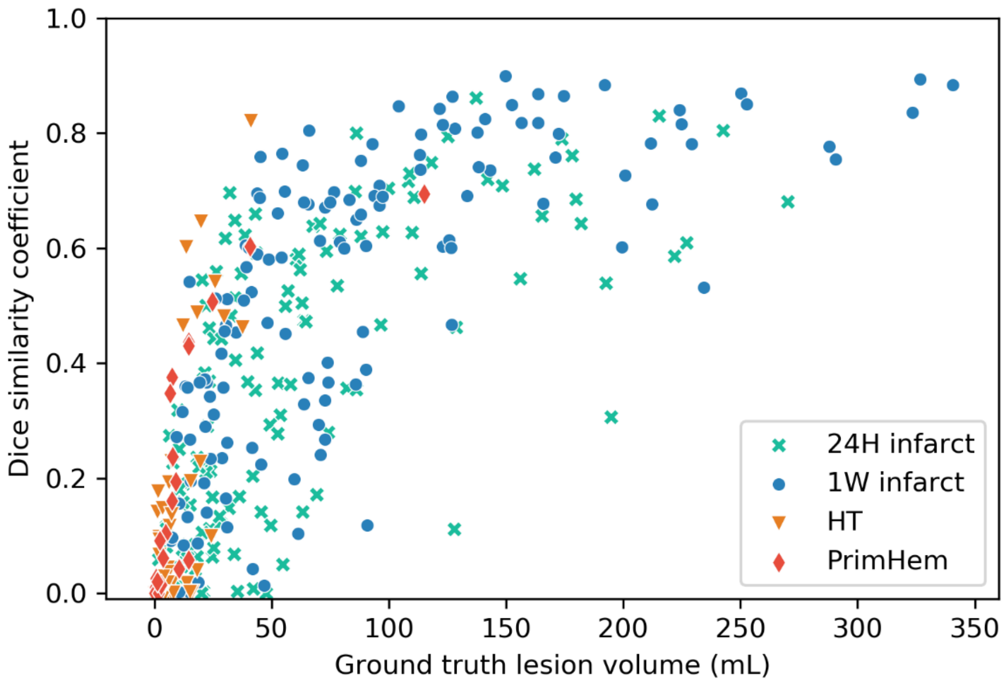



**Figure E.1: Dice similarity coefficient by lesion volume for the FU2BL-GAN.**

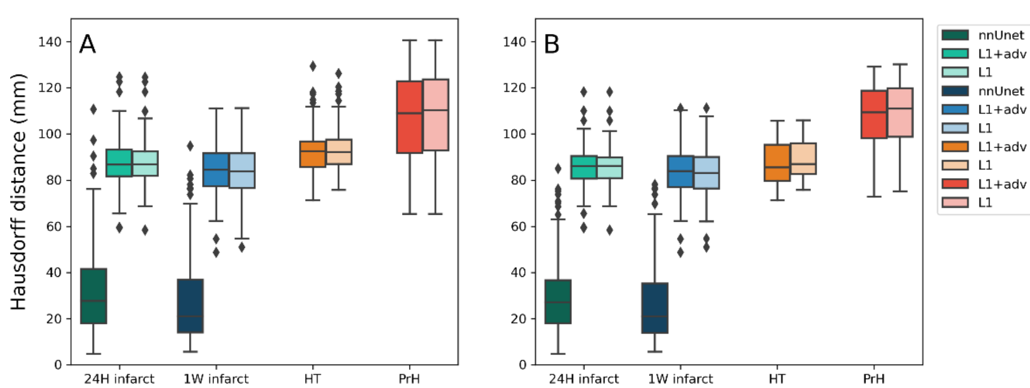



**Figure E.2: Hausdorff distance in mm.** A: considering all data. B: considering only lesions >10mL.

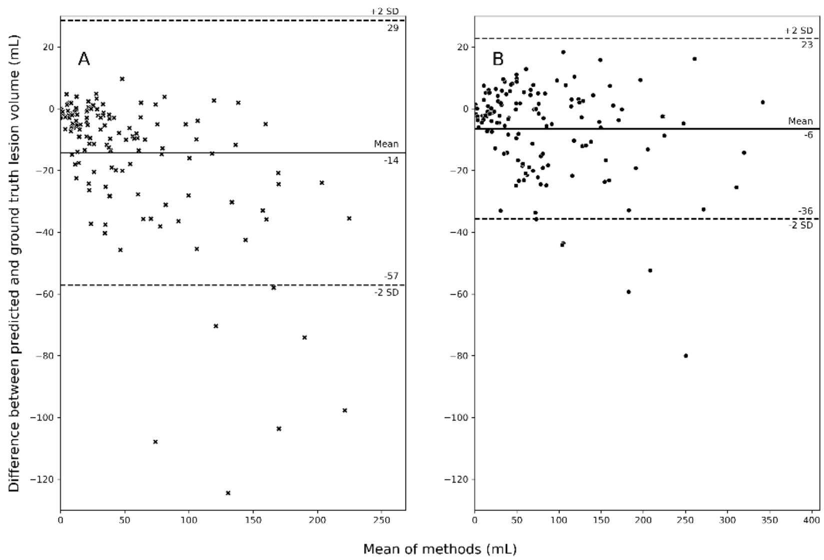



**Figure E.3: Bland-Altman plot of the supervised nnUnet approach.** A: 24 hour infarct lesion. B: 1 week infarct lesion.