Deep Scatter Estimation in PET: Fast Scatter Correction Using a Convolutional Neural Network

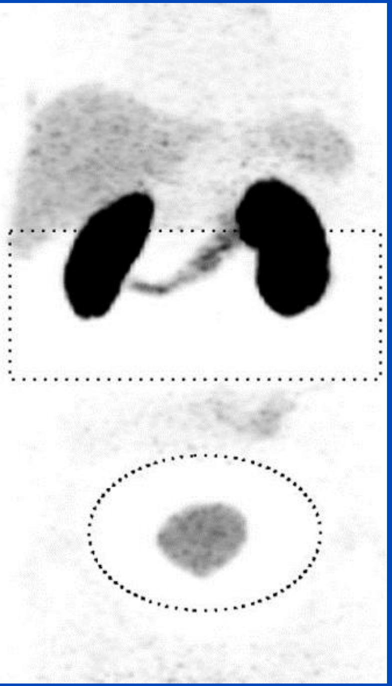
Yannick Berker[1], Joscha Maier[1,2], and Marc Kachelrieß[1,2]

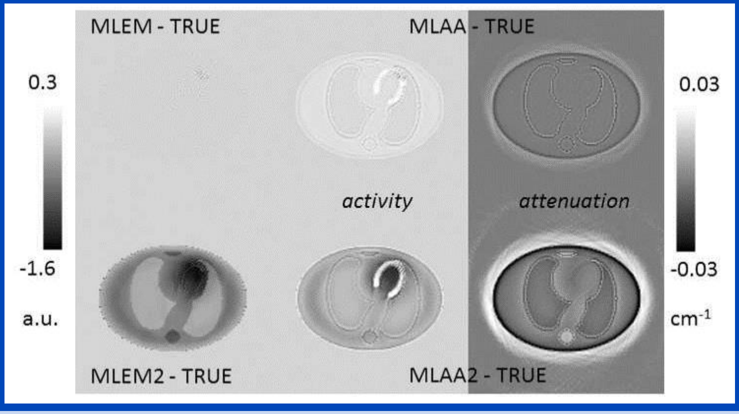
[1]German Cancer Research Center (DKFZ), Heidelberg, Germany

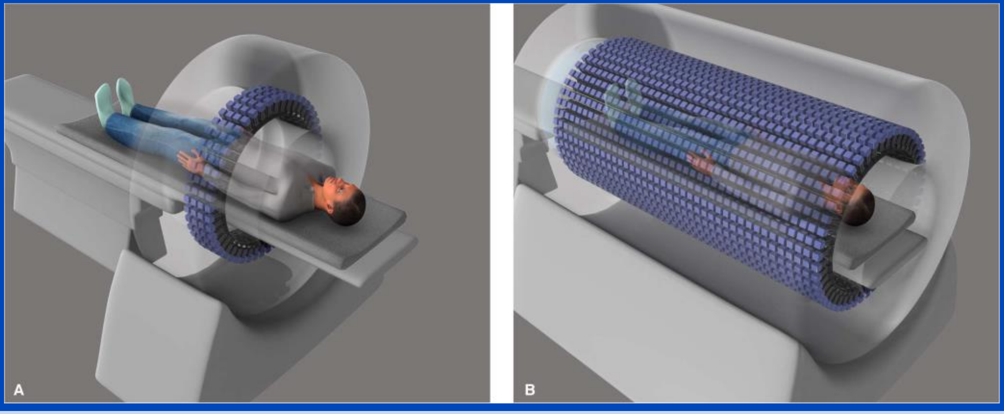
[2]Heidelberg University, Heidelberg, Germany

Scatter-Sensitive PET Applications

- Highly-specific PET tracers[1]
  - Halo effect with ^{68}Ga-PSMA
- Joint estimation[2,3]
  - Unknown radiotracer and attenuation
- Long-axial-FOV PET scanners[4]
  - Need for fast whole-body scatter simulation







[1] Heußer et al. PLoS ONE. 2017;12(8):e0183329.
[2] Heußer et al. IEEE Trans Nucl Sci. 2016;63(5):2443-51.
[3] Nuyts et al. IEEE Trans Rad Plasma Med Sci. 2018;2(4):273-8.
[4] Cherry et al. Sci Transl Med. 2017;9(381):eaaf6169.

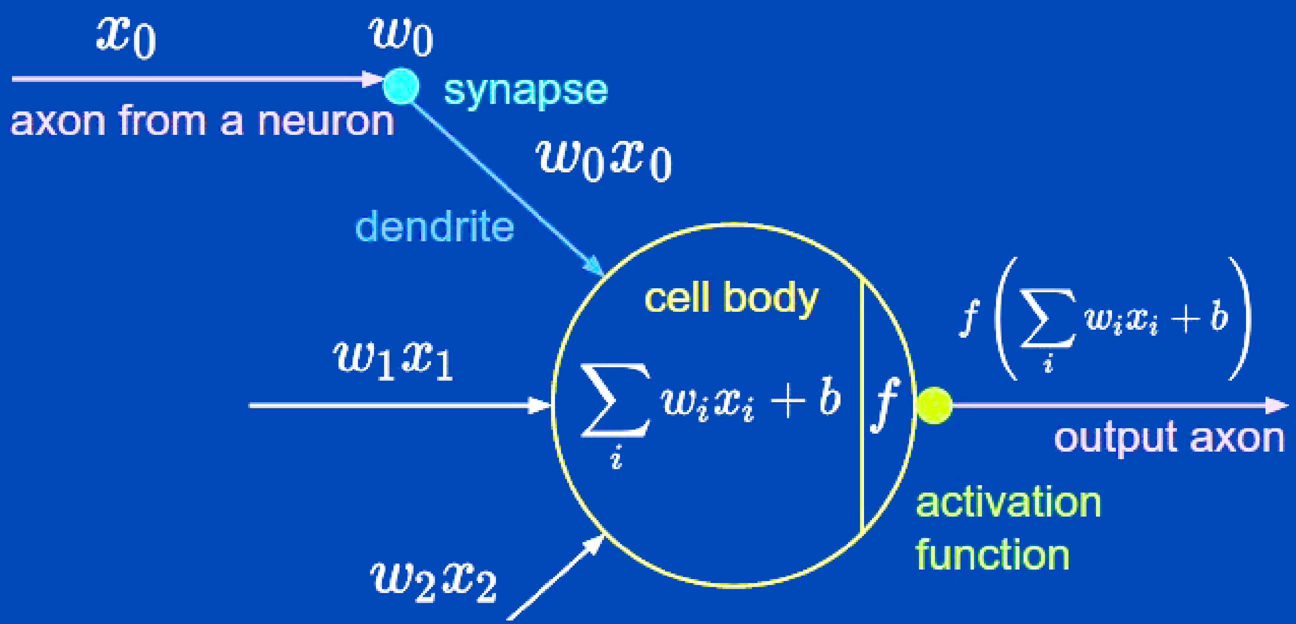
Problem, Aim, Outline

- Monte Carlo scatter simulation (MCSS) is slow
- Single scatter simulation (SSS) is error-prone

➢ **Fast** and **accurate scatter correction** for clinical PET using a deep convolutional neural network (CNN)

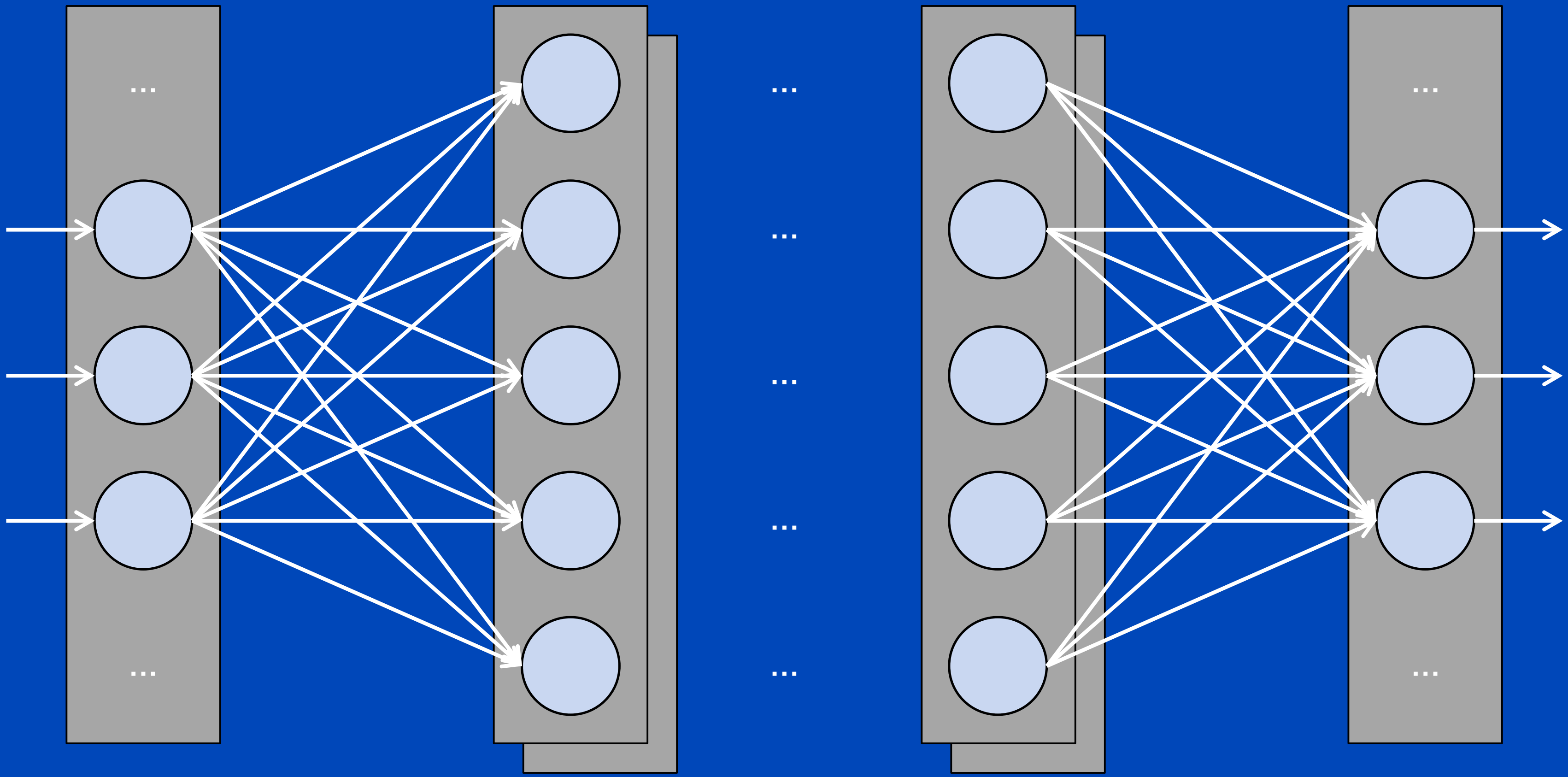
- Background
  - Convolutional neural networks
  - Previous work (CT and PET)
- Deep Scatter Estimation (DSE) in PET

Artificial Neuron[1]



- **Nonlinear activation function** f
- **Multiple inputs, linearly combined**
- **Trainable weights** w_i **and bias** b
- **Supervised learning: adapt parameters to in-/output**

Convolutional Neural Networks



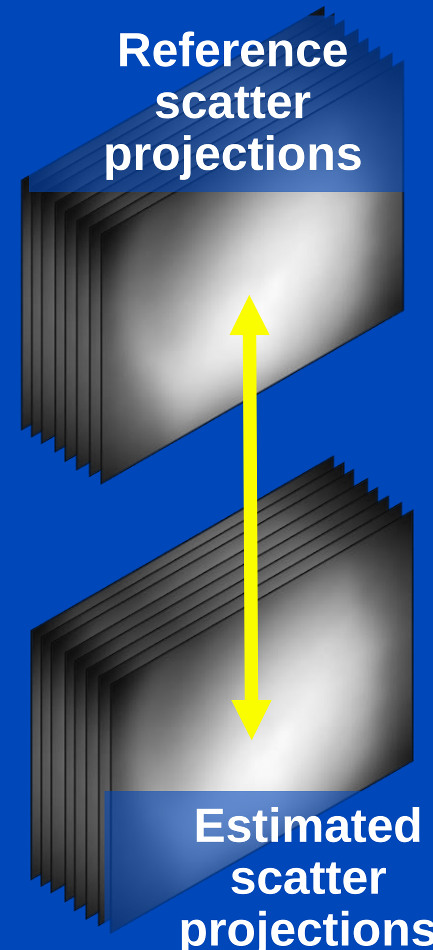
- **Feed-forward:** **no loops**
- **Convolutional** **vs. fully-connected layers of neurons**

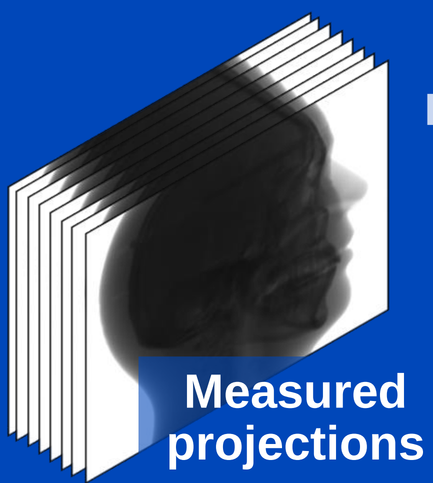
Advantages of CNNs

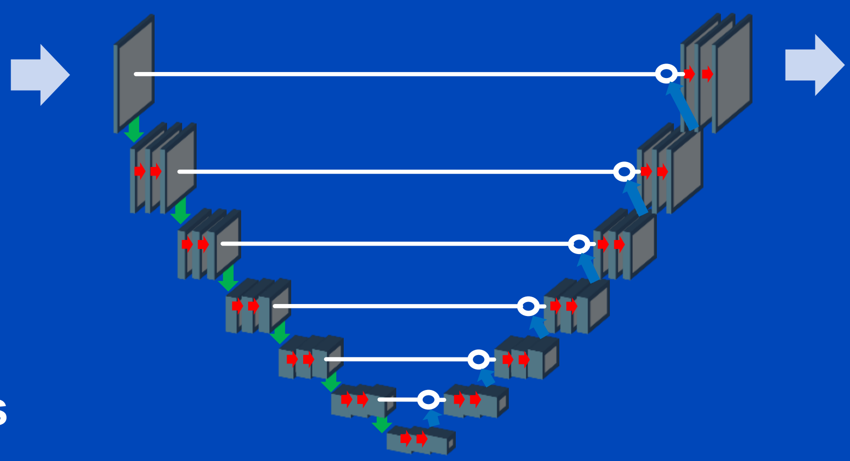
- Once trained, CNNs are fast
  - Potential for clinical PET
- CNNs are versatile
  - Can serve multiple applications
  - Learn from training data
  - General-purpose tools
- Main efforts
  - Definition of network structure
  - Generation of training data

Deep Scatter Estimation in CT

- A 2-D CNN to **estimate scatter** from **scatter-contaminated** projections[1]
  - Trained using measurements and reference
  - Applied to individual projections
  - ➢ Real-time performance for cone-beam CT







[1] Maier J, Eulig E, Vöth T, Knaup M, Kuntz J, Sawall S, Kachelrieß M. Med Phys. 2018. https://doi.org/10.1002/mp.13274
Also compare Hansen DC, Landry G, Kamp F, Li M, Belka C Parodi K, Kurz C. Med Phys. 2018;45(11):4916-26.

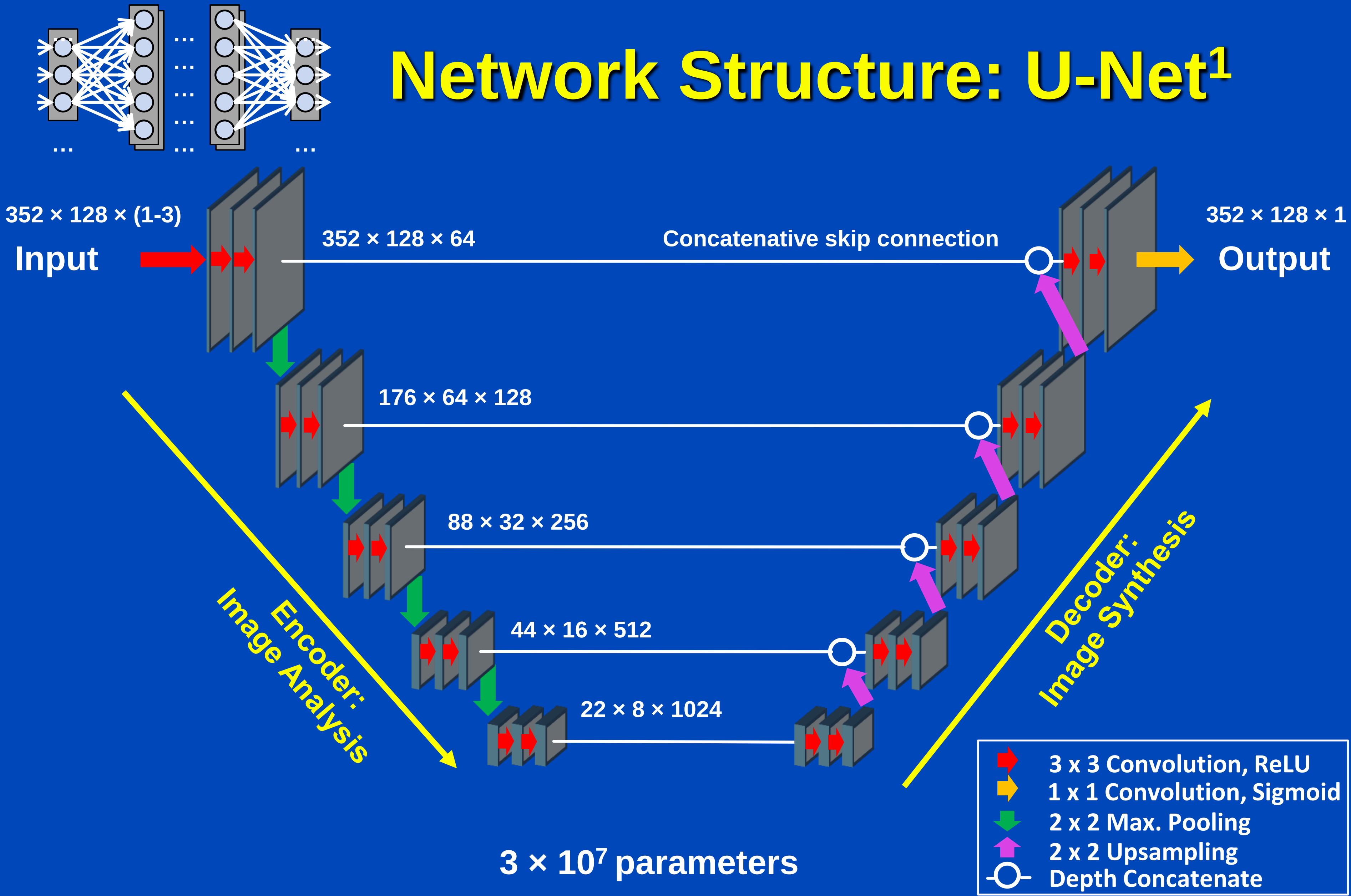
Deep Scatter Estimation in PET

- Previous work[1]
  - 2-D “neural network that predicts total scatters from emission and attenuation data” (for views)
  - Monte Carlo simulations of 13/1 phantoms (training/validation)
  - Showed promise “but needs more work”

➢ **Proof of concept:** single scatter simulation using a deep convolutional neural network (CNN)
  - Network structure
  - Human training data
  - Speed and accuracy

[1] Qian H, Rui X, De Man B. IEEE Nucl Sci Symp Med Imaging Conf. 2017;M04-1.
Also compare Yang J, Park D, Wang ZJ, Seo Y. IEEE Nucl Sci Symp Med Imaging Conf. 2018;M14-360.

Network Structure: U-Net[1]

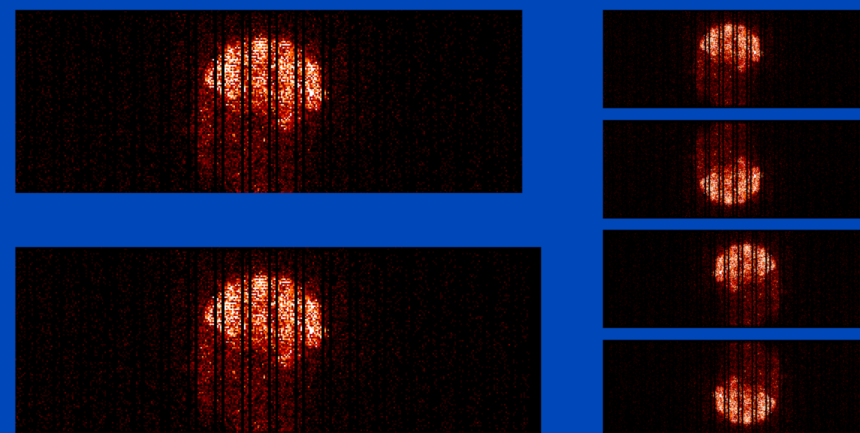


3×10^7 parameters

[1] Ronneberger O, Fischer P, Brox T. MICCAI. 2015:234-41.

Patient Data

- **20 patients**
  - FDG, Siemens Biograph mMR
  - 2-6 bed positions, 252 views
- **Zero-padding**
  - 344 × 127 → 352 × 128 pixels



- **4x data augmentation**
  - Horizontal and vertical flipping
  - 71,568 views
- **3 input features** Prompts, 1/ACF, log ACF
- **1 output feature** SSS (readily available, unlike MCSS)

ACF, attenuation correction factor. SSS, single scatter simulation, MCSS, Monte Carlo scatter simulation

Implementation

- 80/20 split for training/validation
  - 57/14 bed positions
- Poisson loss function
  - $\sum_i (DSE_i - SSS_i \cdot \log(DSE_i + \epsilon))$
- Adam optimizer
  - batch size 4, learning rate 10^{-4} (with reduction), 20 epochs

- TensorFlow w/ Keras 1.12.0, Python 3.6.7
- Intel Xeon E5-2667 v4 (2 x 8 cores, 256 GB)
- NVIDIA Quadro M5000 (2048 cores, 8 GB)

Metrics

- **Normalized Mean Absolute Error**

$$NMAE = \frac{\sum_i |DSE_i - SSS_i|}{\sum_i |SSS_i|}$$

✓ **Unique normalization***
✓ **Percentages**
✓ **FOV independent**
✓ **Stable**

- **2 NRMSEs of PET body areas: -20% or +50%**

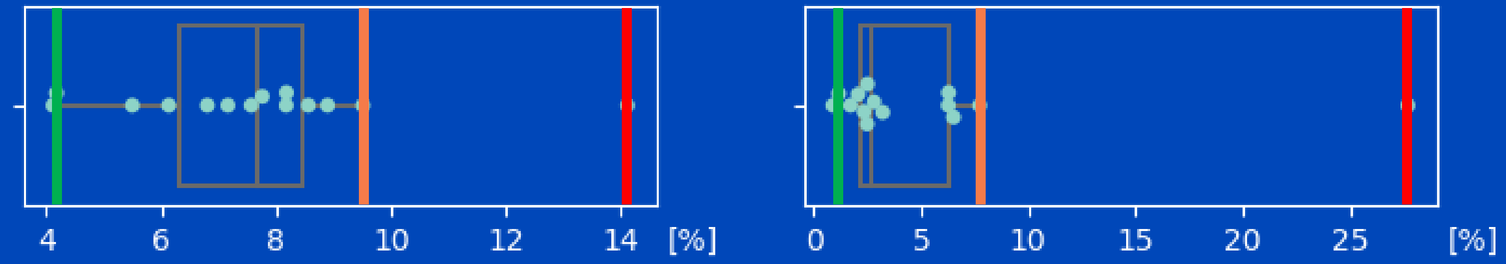
Results: Speed

- **Training duration**
  - **32 hours (57,456 training views, 20 epochs)**
  - **Scales linearly with size of training data**
- **Prediction duration**
  - **21 ms per view**
  - **5.3 s per bed position**
  - **< 30 s for 5 bed positions**
  - **SSS: 3.5 minutes (log files)**

SSS, single scatter simulation

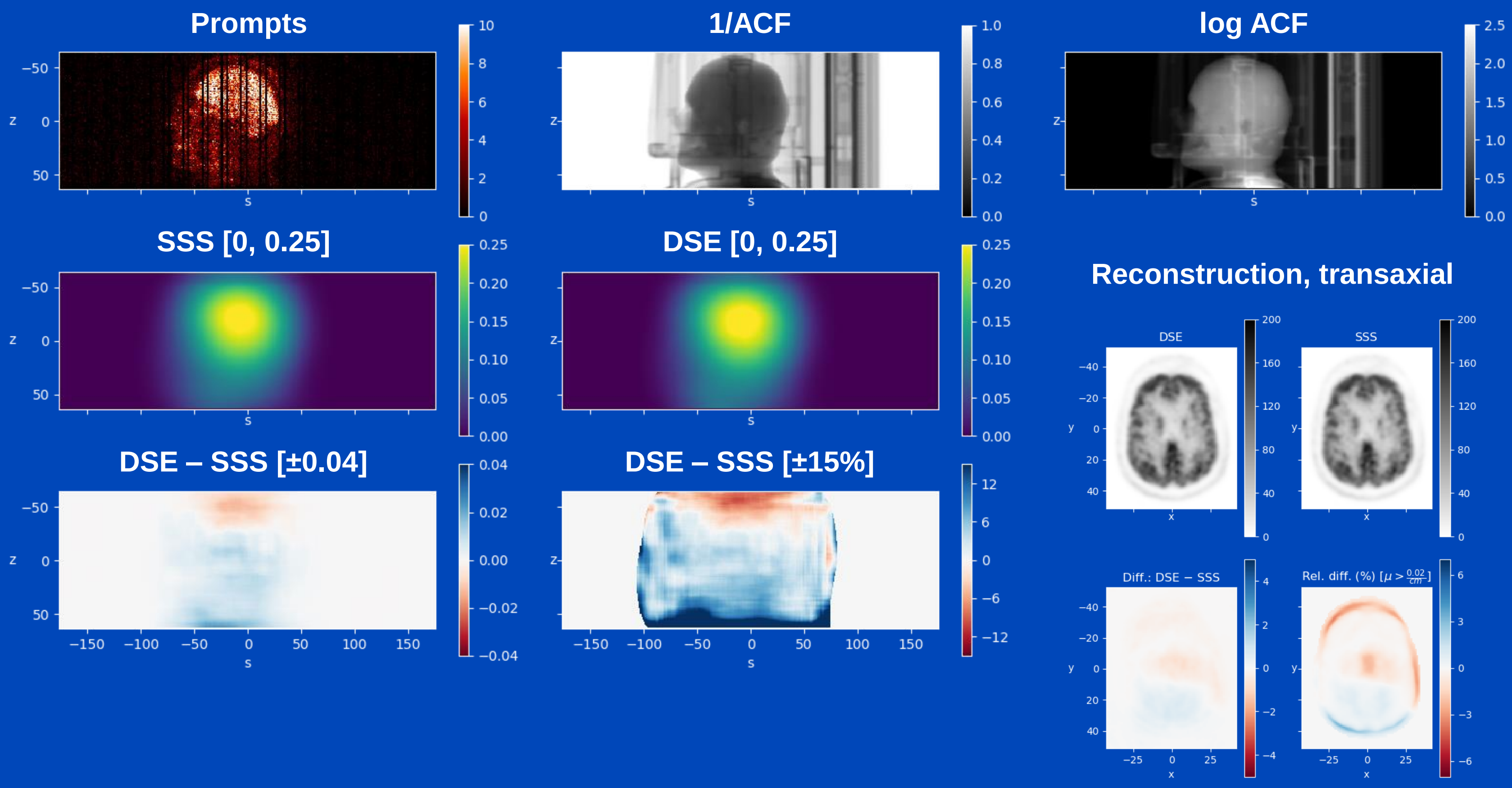
Results: Accuracy

NMAE	Scatter	Recon
Mean/Std	7.1 ± 1.7 %	3.6 ± 2.2 %
Range	4 – 10 %	1 – 8 %
Outlier	14 %	28 %



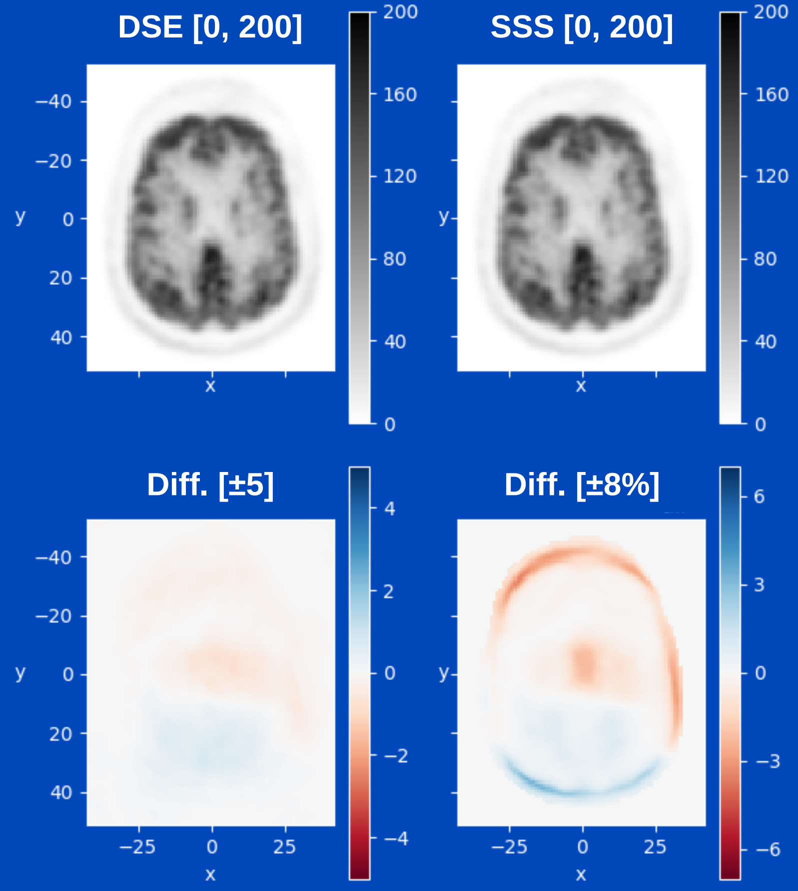
NMAE, normalized mean absolute error (per bed position)

Results: Best Case



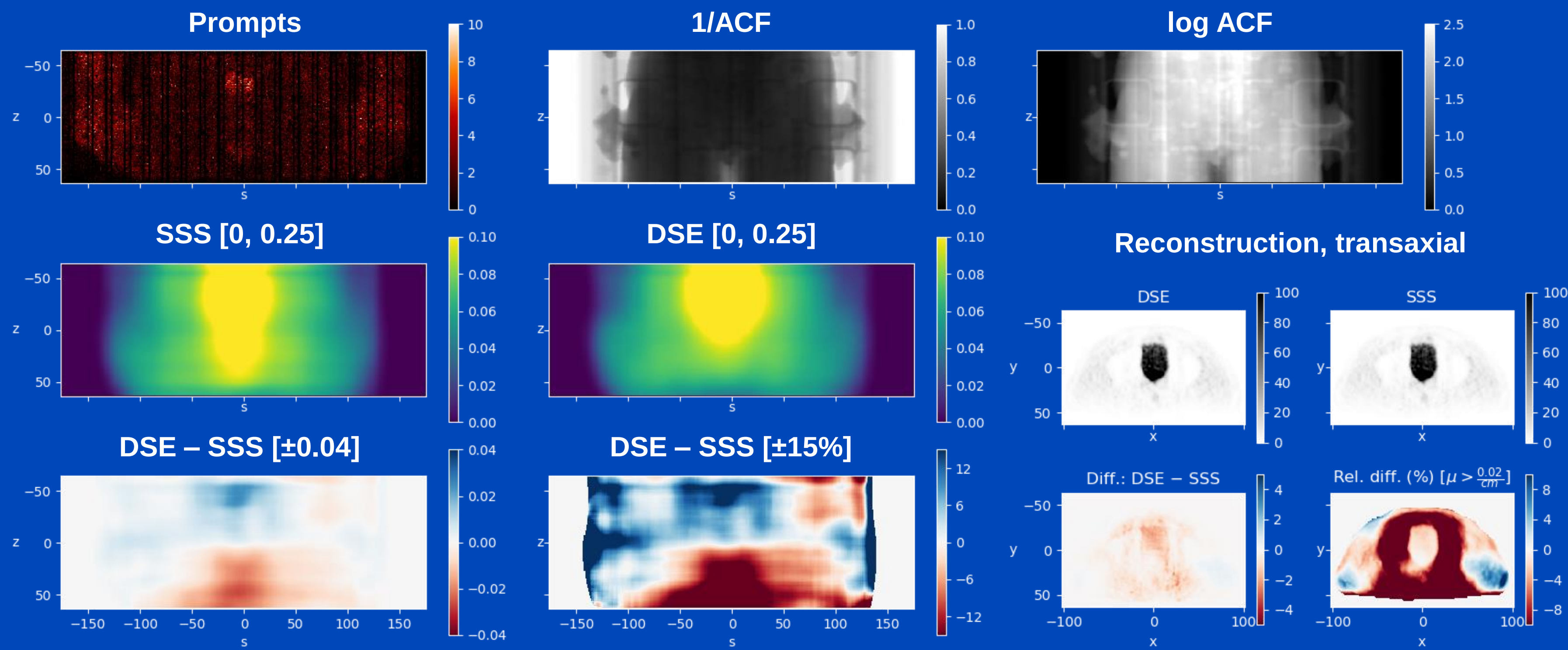
- **Best case:** **brain bed position**

Results: Best Case



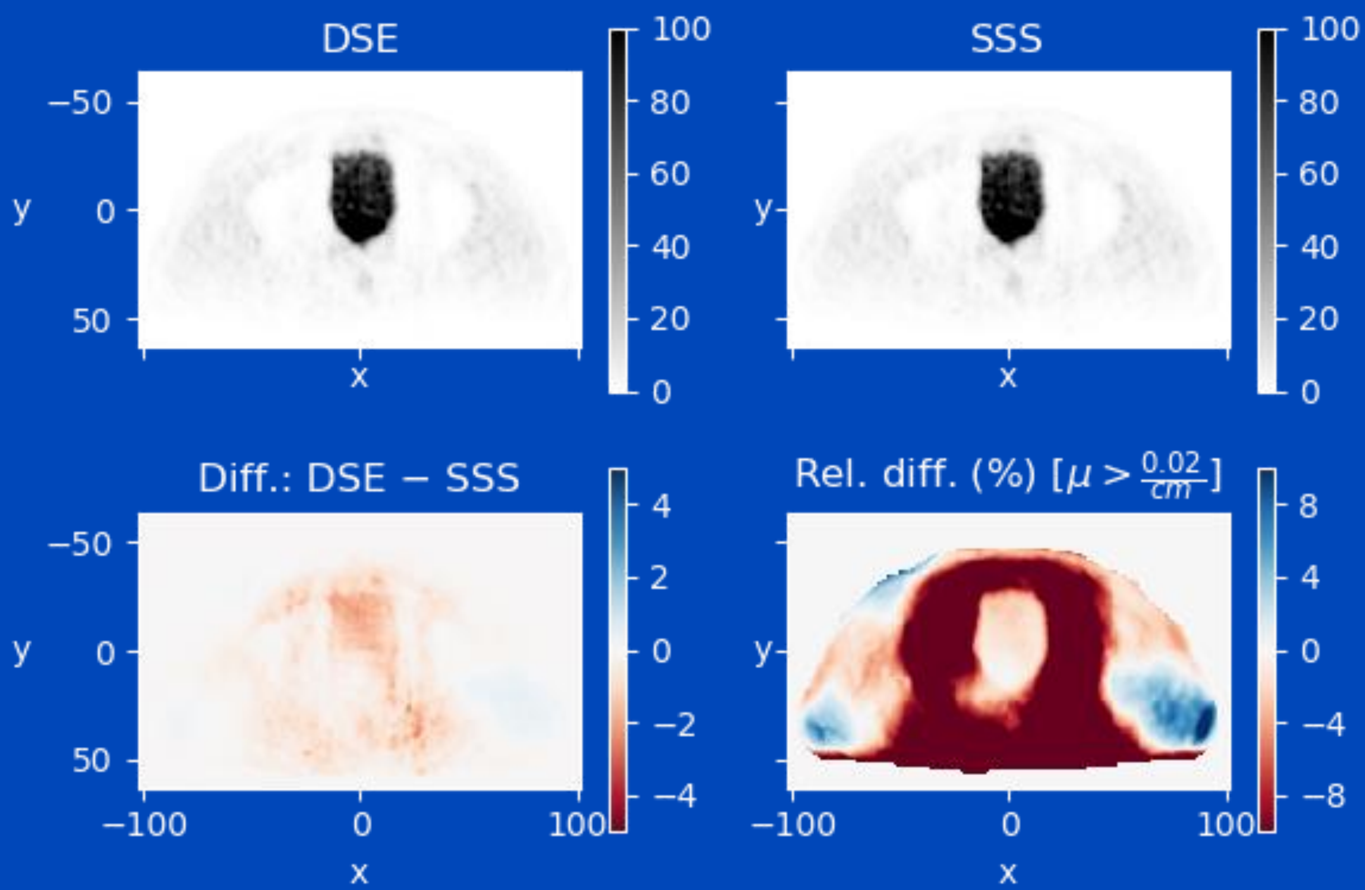
Bed position e7876f, NMAE: 4.17 % (scatter), 1.18 % (recon)
Reconstruction, transaxial (a.u.), static

Results: Worst Case

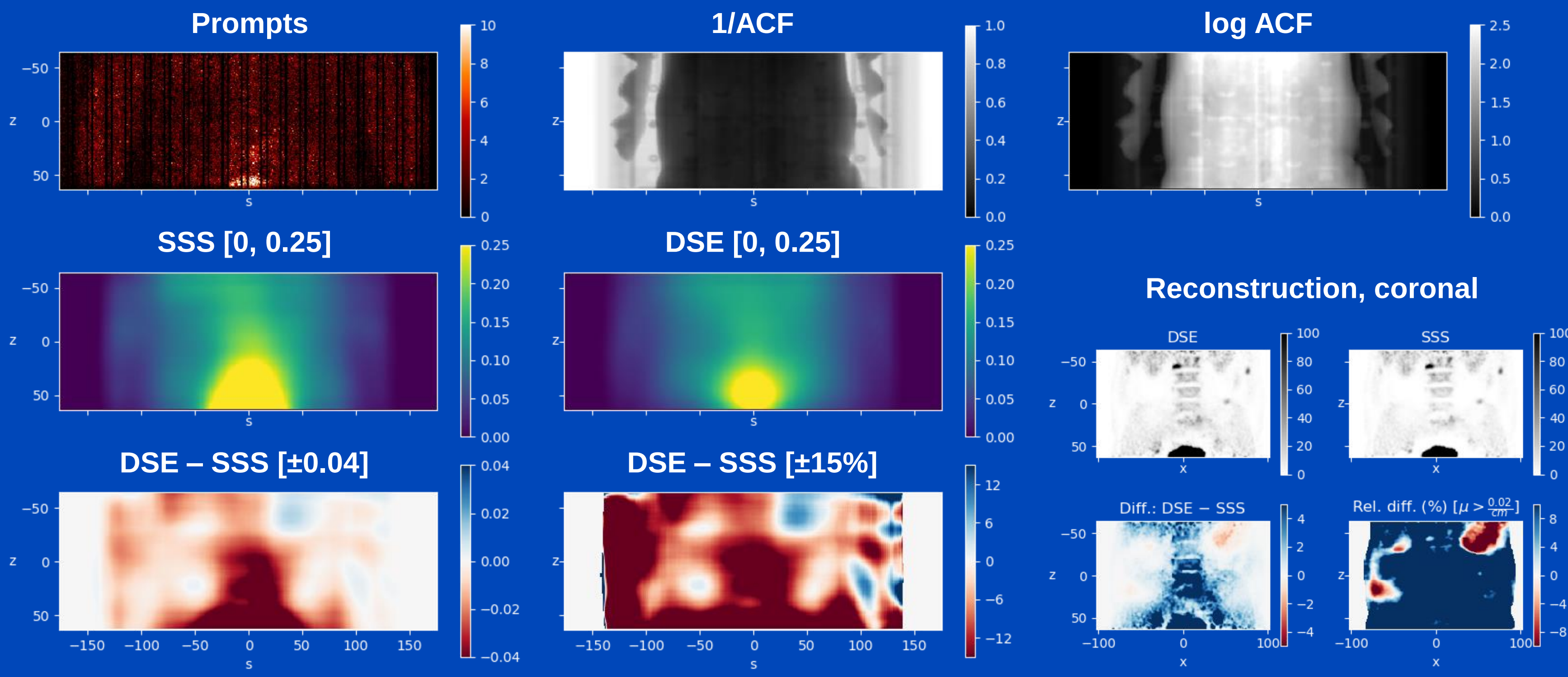


- **Worst case: filled bladder inside the FOV**

Results: Worst Case

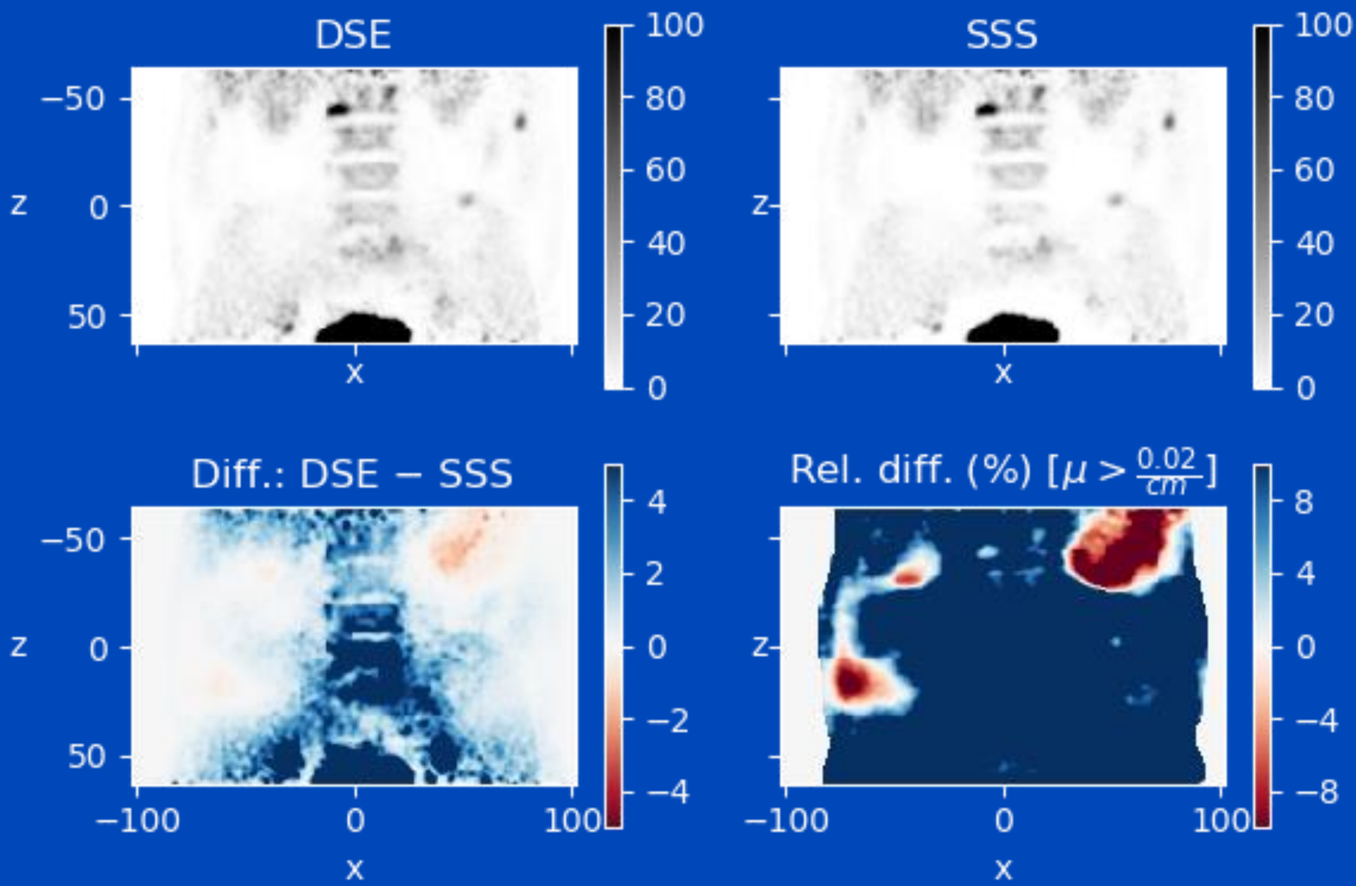


Results: Outlier



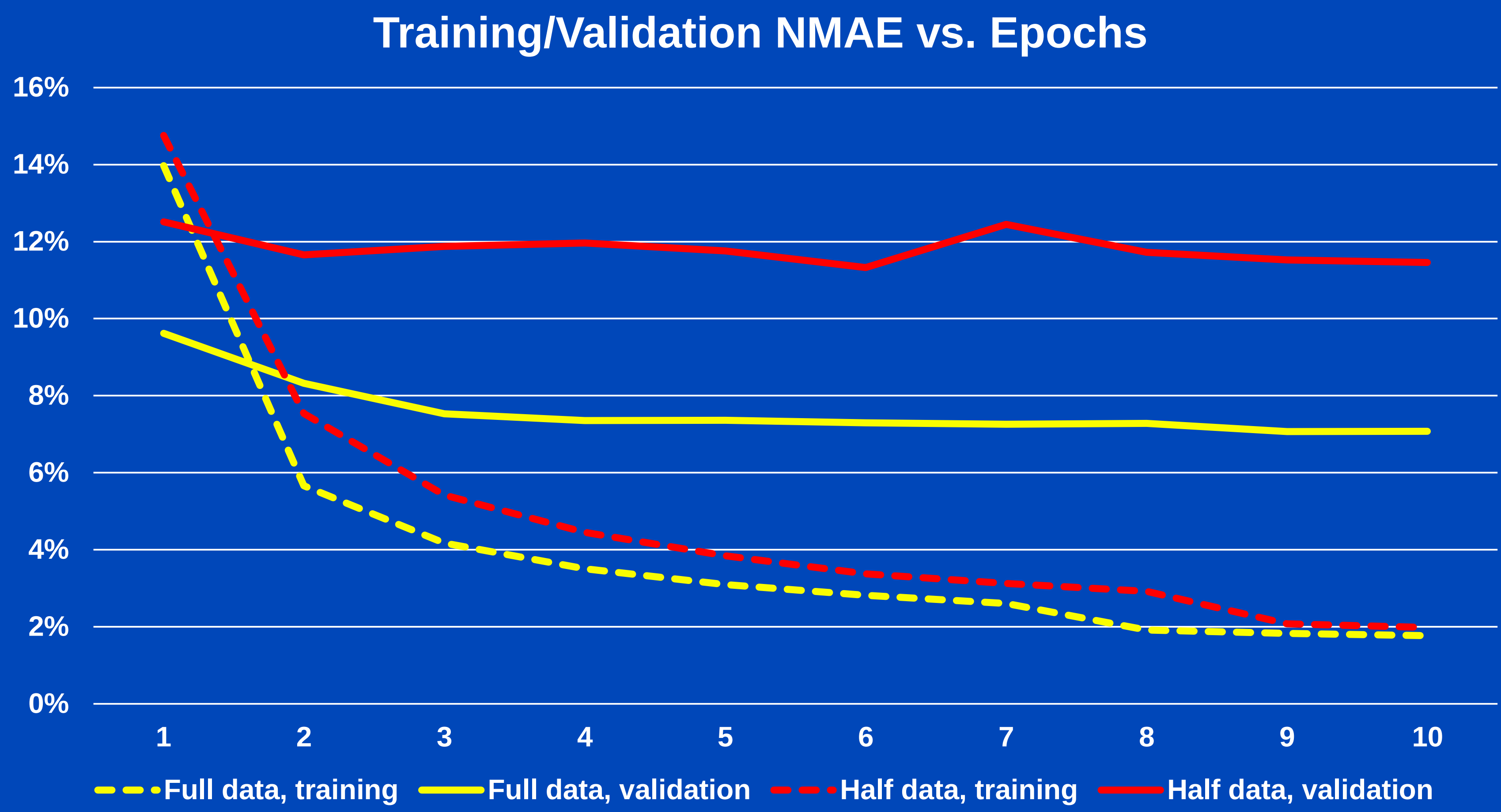
- **Outlier:** filled bladder **extending outside** the FOV

Results: Outlier



Bed position dcfb3b, NMAE: 14.11 % (scatter), 27.70 % (recon)
Reconstruction, coronal (a.u.), static

Number of training data



NMAE, normalized mean absolute error

Conclusion

- A U-Net CNN can reproduce Siemens SSS non-iteratively, with good accuracy, in 5 seconds.
- More training data may be needed
  - Cross-bed-position data augmentation (→ whole-body scatter simulation)
- Prostate scans: improvements necessary
  - Organ-specific training
- Aim: MC-DSE trained for Monte Carlo scatter

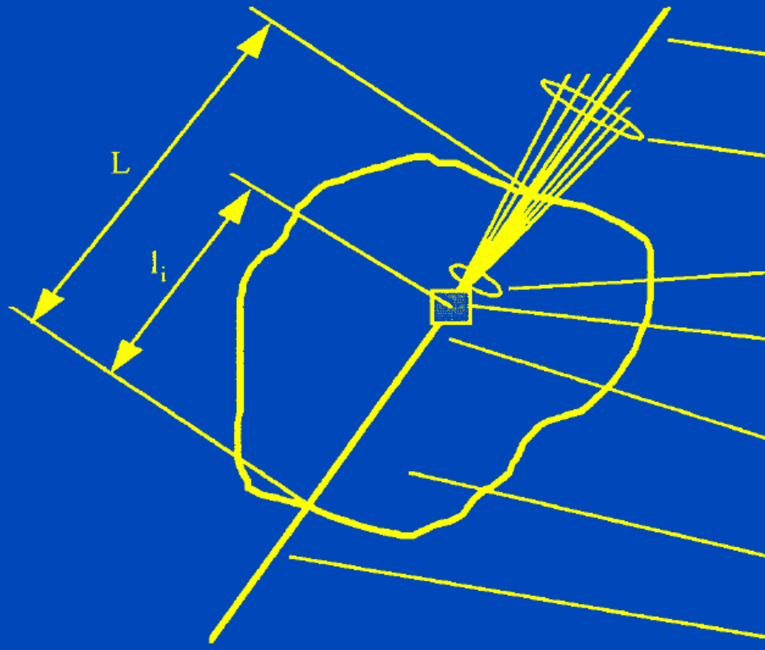
Thank You!

This presentation will soon be available at www.dkfz.de/ct.

Supported by a DKFZ Postdoc fellowship – also apply for a DKFZ PhD fellowship.
Parts of the reconstruction software were provided by RayConStruct® GmbH, Nürnberg, Germany.

Choice of Input Features

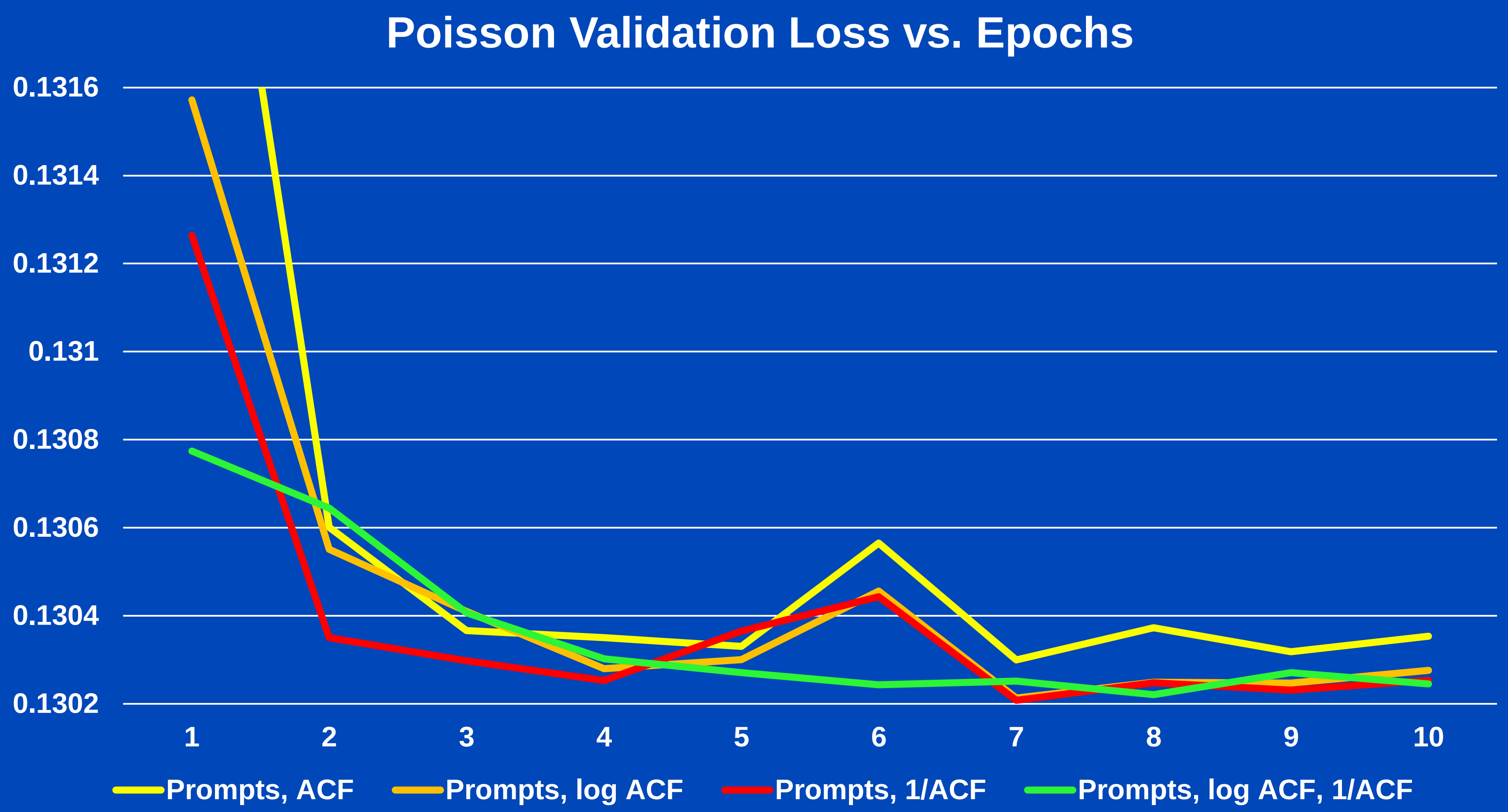
- Small scattering angles
- $p = \int \mu \, \mathrm{d}x$, $\mathrm{ACF} = \exp(p)$



- X-ray CT[1] ("pep model")
  - Scatter $(\mathrm{I}_0 \cdot p \cdot \exp(-p)\,) ** K$
  - Measure $p' = \log(I/I_0) \approx p$
  - CNN input[2] $p' \cdot \exp(-p')$
- PET ("clever name here")
  - Scatter $\left(\int \lambda \, dx \cdot p \cdot \exp(-p)\right) ** K$
  - Measure $\text{prompts} \approx \int \lambda \, dx$
  - CNN input prompts, $p = \log \mathrm{ACF}$, $\exp(-p) = 1/\mathrm{ACF}$

[1] Ohnesorge B, Flohr T, Klingenbeck-Regn K. Eur Radiol. 2010;9(3):563-9.
[2] Maier J, et ind, Kachelrieß M. Med Phys. 2018. https://doi.org/10.1002/mp.13274

Impact of Input Features



ACF, attenuation correction factor