**University of Stuttgart**
Institute of Interfacial Process Engineering and Plasma Technology
Pfaffenwaldring 31, 70569 Stuttgart, Germany
<u>Christos Vagkidis</u>, E. Devlaminck, A. Köhn-Seemann, S. Merli, M. Ramisch, A. Schulz, G. Tovar

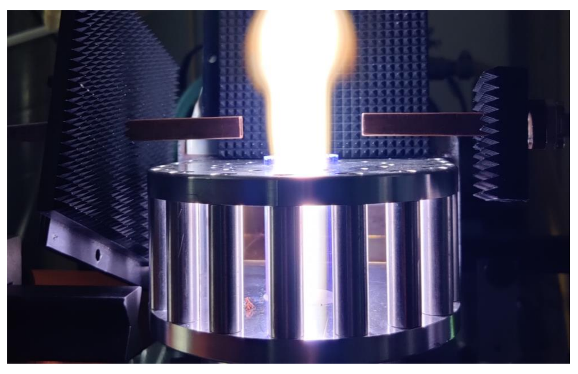
# A machine learning approach to the inverse scattering problem

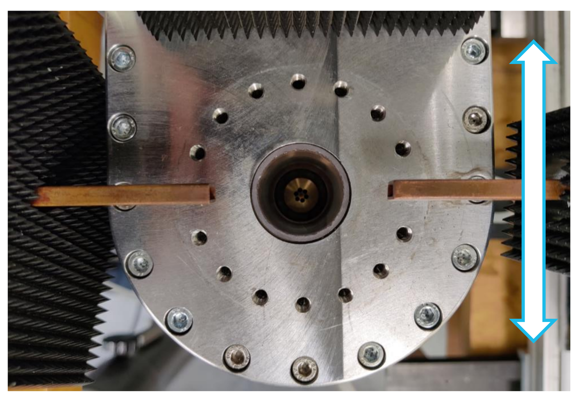
## Introduction

Single-chord interferometry is limited by measuring only the line-integrated density of a non-uniform plasma. Here we propose the use of a deep neural network (NN) to solve the inverse scattering problem and retrieve a 2D density profile of an atmospheric plasma torch. The NN is trained on data obtained from 3D simulations executed with the COMSOL Multiphysics software. A high-frequency microwave is penetrating the plasma and is scattered by it. An extensive variation of the electron density profile in the simulations leads to different beam scattering profiles, which are used to train the NN. Ideally, the NN is capable of linking the scattering profile to the plasma density.

## The plasma torch

- Microwave generated atmospheric plasma
- Powered by 2.45 GHz microwave source (nominal heating power up to 3 kW [1])
- 140 GHz microwave used for diagnostics
- Single-chord interferometry yields line-integrated density
- Movable receiving antenna measures spatial beam power (scattered)
- Scattering profile depends on plasma profile

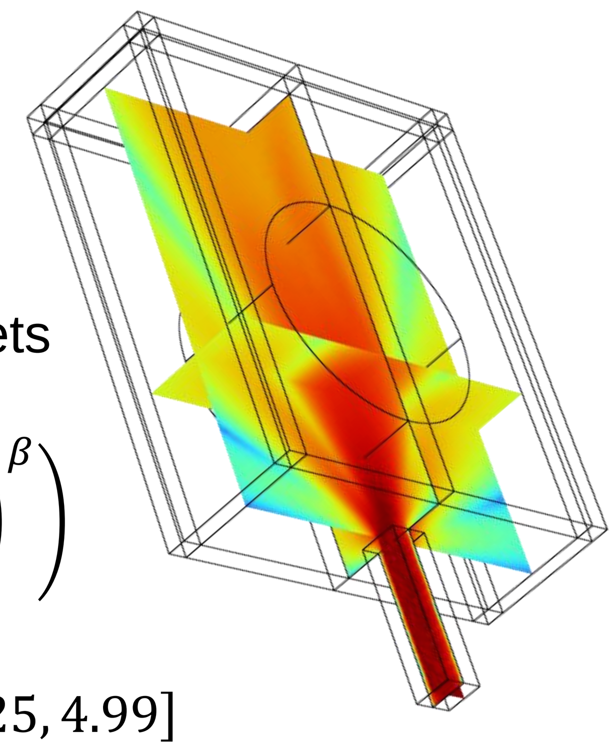




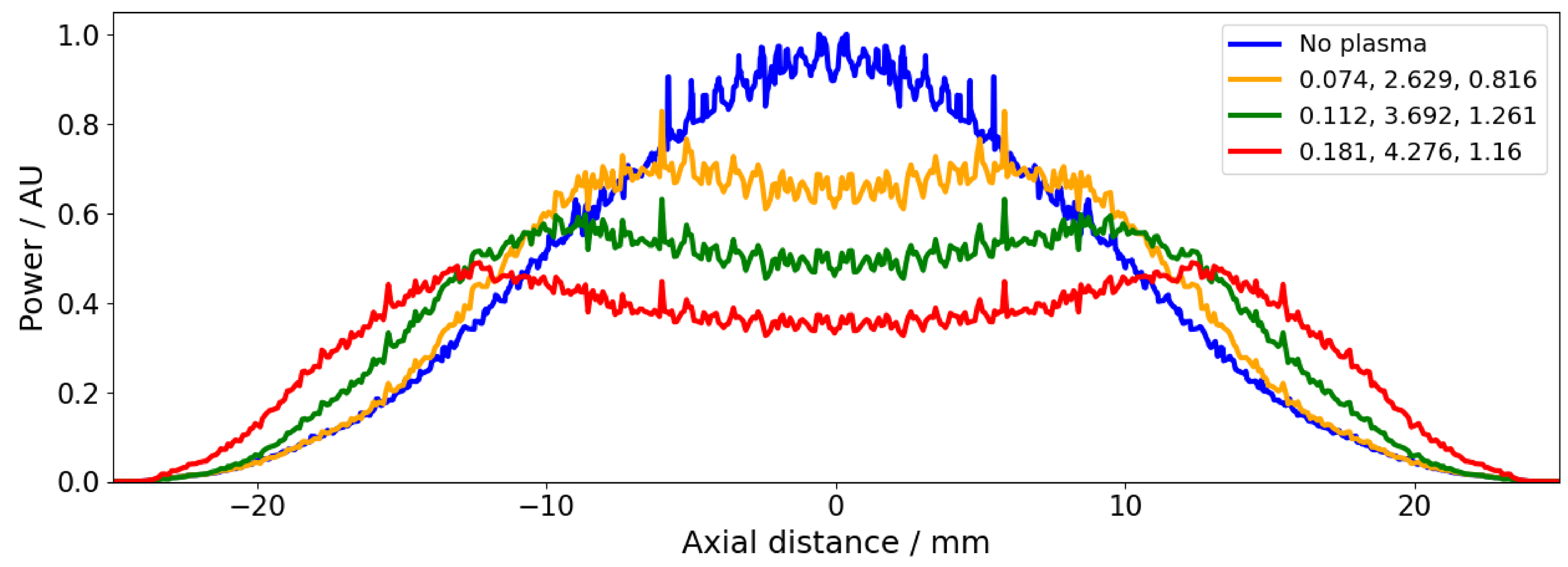
## Data set for NN

- 1000 beam scattering profiles
- Profiles obtained from 3D full-wave simulations with COMSOL Multiphysics
- Data split into 80% training and 20% test sets
- Plasma profile used:

$$n_e(x,y) = \frac{n_e}{n_c} \exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)^{\beta}\right)$$

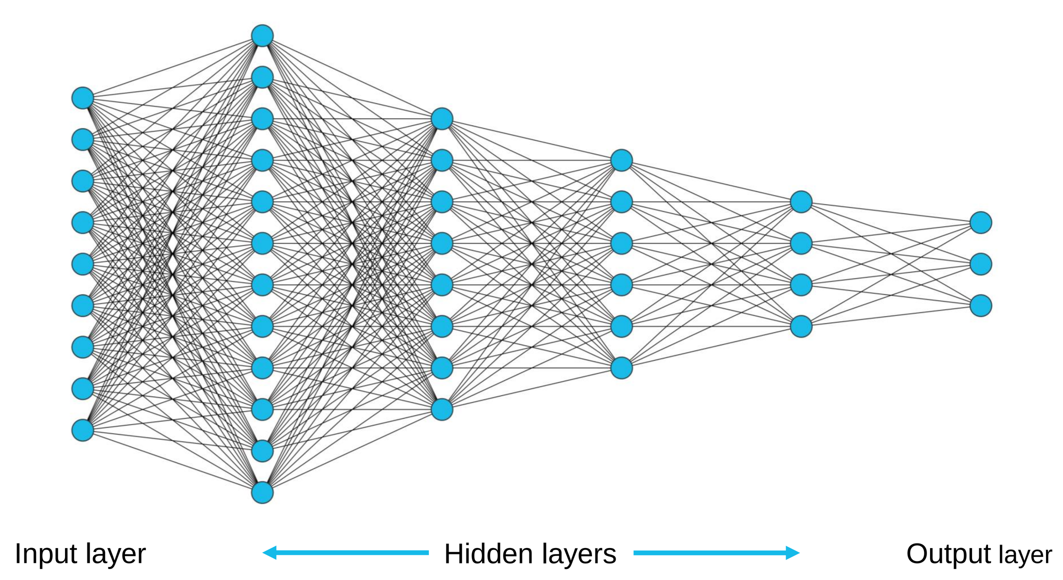


- Parameter range:
  - $n_e/n_c \in [0.05, 0.2]$
  - $\sigma = \sigma_x = \sigma_y \in [1.25, 4.99]$
  - $\beta \in [0.3, 2.5]$




## Architecture of NN

- Functional API model built with Keras [2]
- Input layer: Gaussian mixture parametrization of scattering profile (9 parameters: 3 x peak, mean, standard deviation)
- NN maps input layer to plasma profile (3 parameters as output layer)

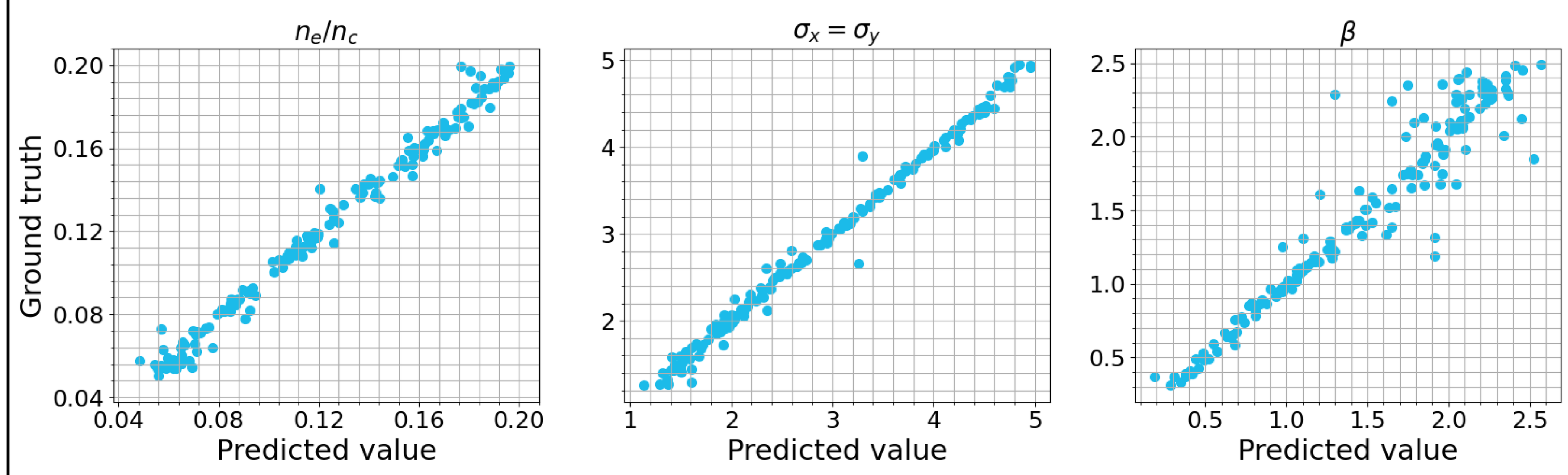


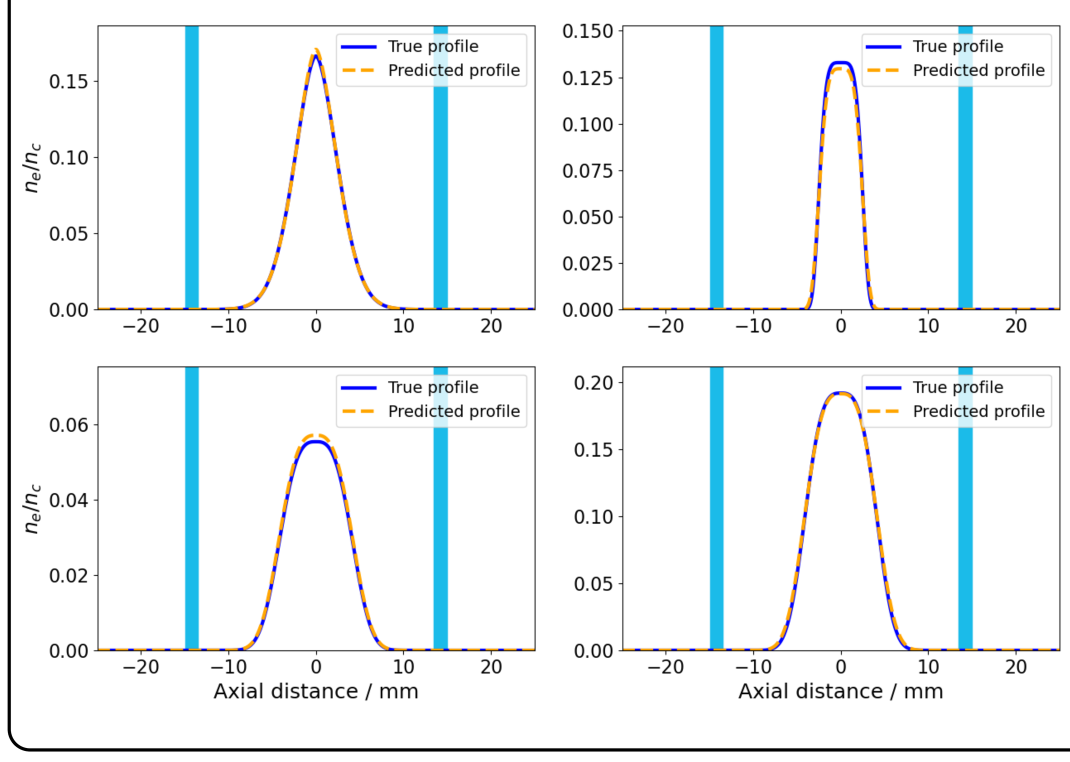

- Current NN state
  - 4 hidden layers
  - 184 neurons
  - Activation = *ReLu*
  - Optimizer = Adam
  - Loss = MAE

## References

[1] K. Wiegers *et al*, Chem. Ing. Tech. **94**, 3, 299-308 (2021)
[2] F. Chollet *et al*, Keras, https://keras.io (2015)

## Performance of NN



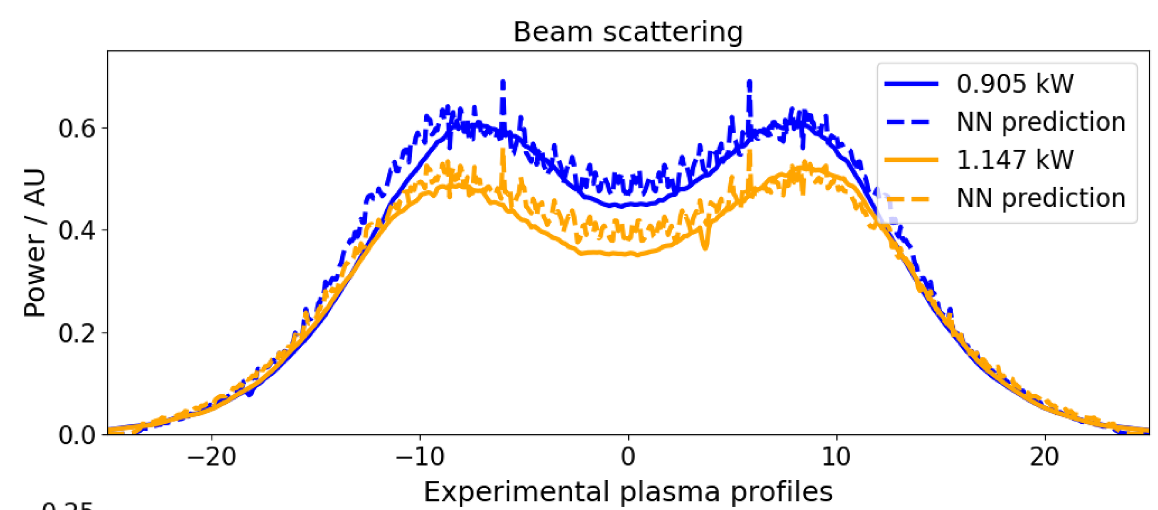



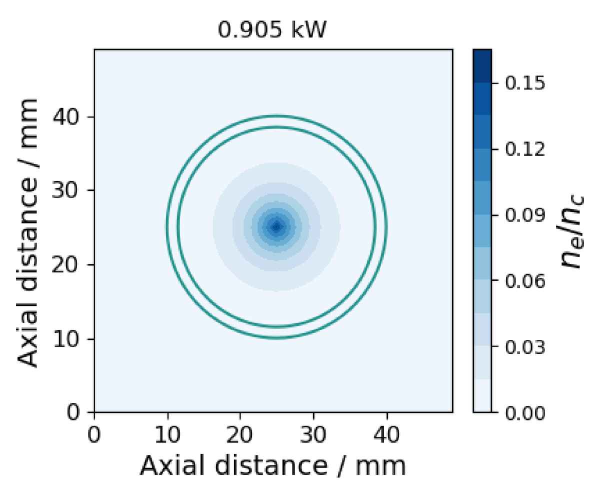

| | MAE | | MAPE (%) | |
|---|---|---|---|---|
| | Train | Test | Train | Test |
| $\frac{n_e}{n_c}$ | 0.057 | 0.079 | 2.42 | 3.52 |
| $\sigma$ | 0.038 | 0.051 | 1.70 | 2.36 |
| $\beta$ | 0.101 | 0.155 | 4.49 | 6.62 |

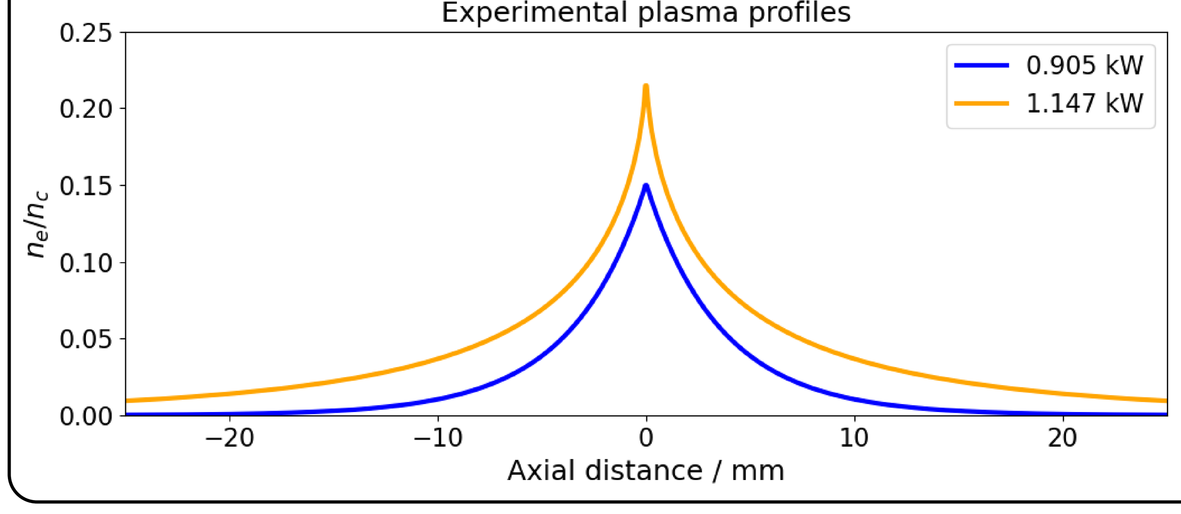
MAE: Mean Absolute error
MAPE: Mean Absolute Percentage Error

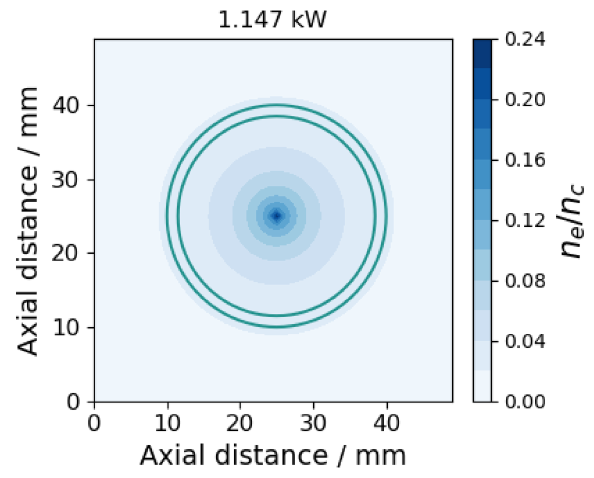
## Experimental prediction

1. Measure experimental beam scattering
2. Feed parametrized scattering profile into NN
3. Predict experimental plasma profile
4. Run simulation with predicted plasma profile
5. Compare simulation result against experimental
6. Repeat for different input heating power




## Summary and outlook

- Functional API model built to predict density profile of plasma torch
- MAPE on $n_e/n_c$ and $\sigma$ < 3% for training data and < 4% for test data
- MAPE on $\beta$ < 4.5% for training data and < 7% for test data
- Network is planned to be trained on more data to enhance performance
- Parameter range will be expanded for better generalization