Machine learning in low temperature plasma science

Tobias Gergs[1,2], Borislav Borislavov[1], Florian Krüger[1,3], and Jan Trieschmann[1]

[1]Brandenburg University of Technology
[2]now with Ruhr University Bochum
[3]now with University of Michigan

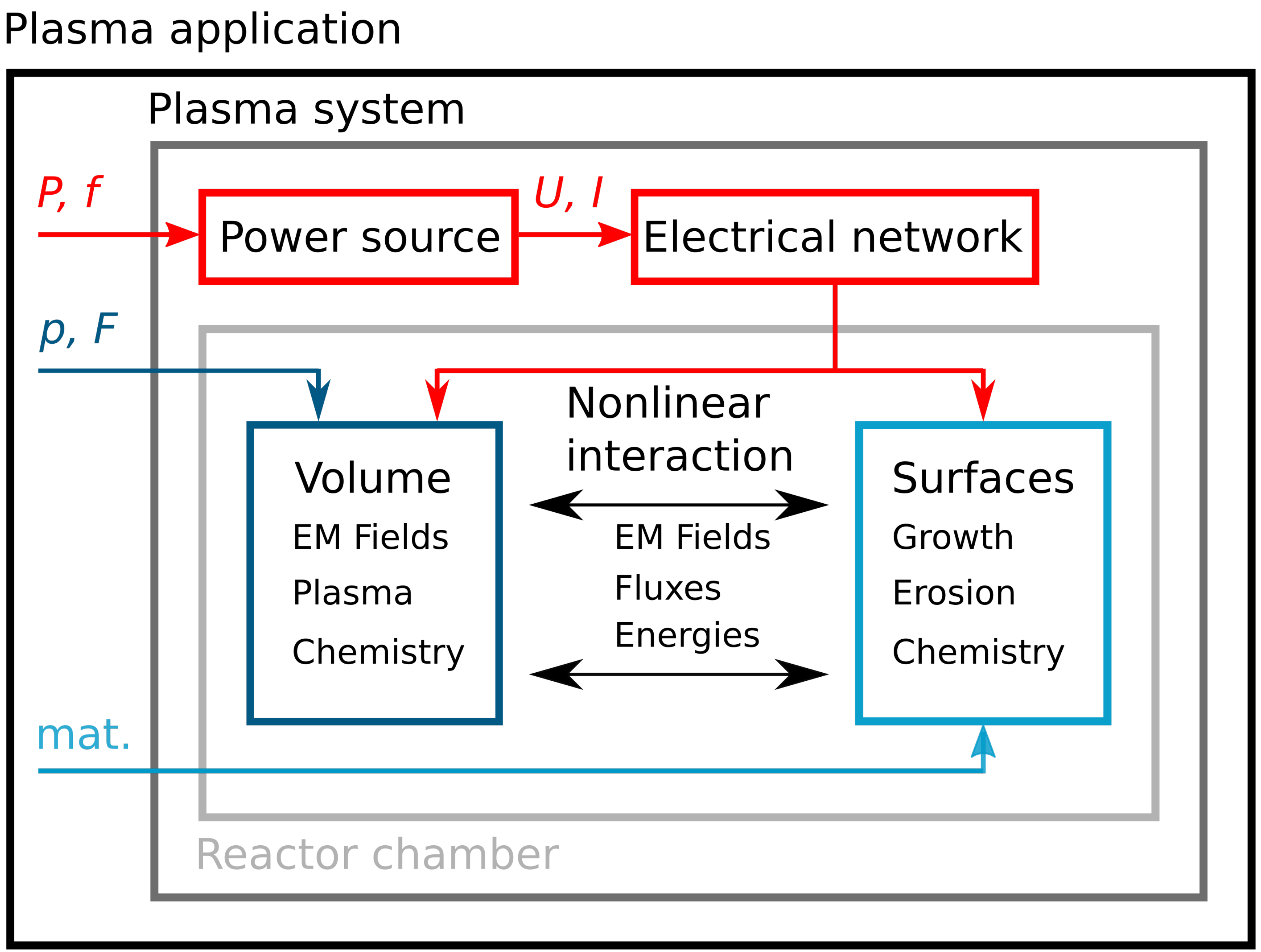
International Online Plasma Seminar

April 8, 2021

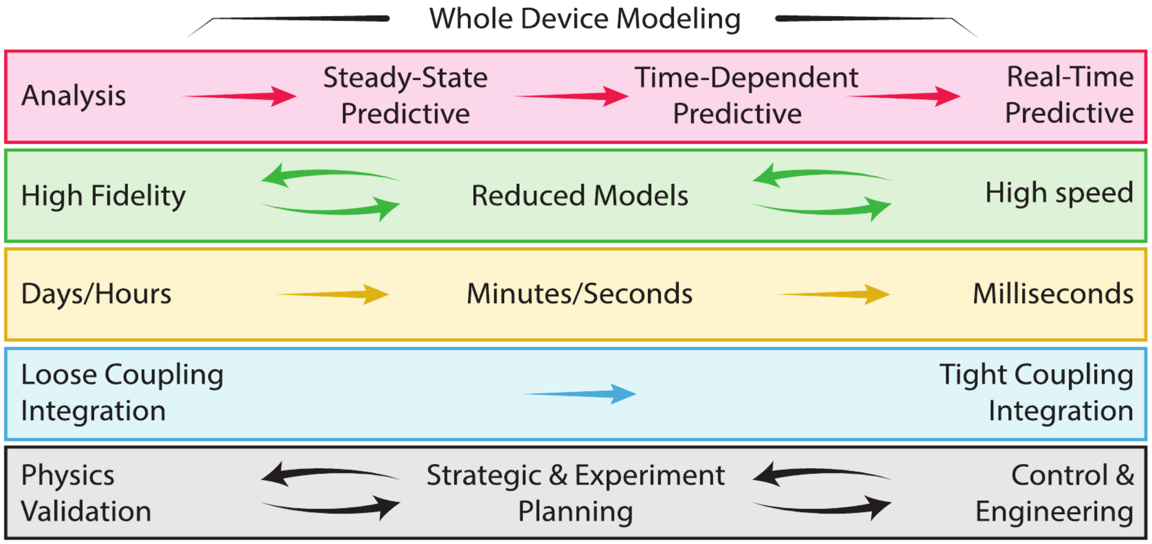
Jan.Trieschmann@b-tu.de
https://www.trieschmann.org

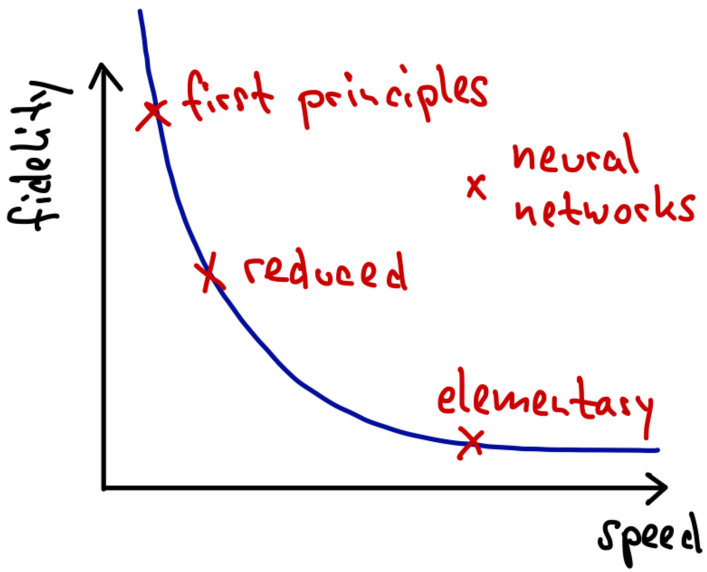
Brandenburg
University of Technology
Cottbus - Senftenberg

Low-temperature plasma systems



Machine learning surrogate models





- Elementary models faster at the cost of physical fidelity
- Nonlinear multi-dimensional regression to high fidelity database
- Neural networks may break fidelity vs speed trade-off

Meneghini et al., IAEA Technical Meeting (2017)

Outline

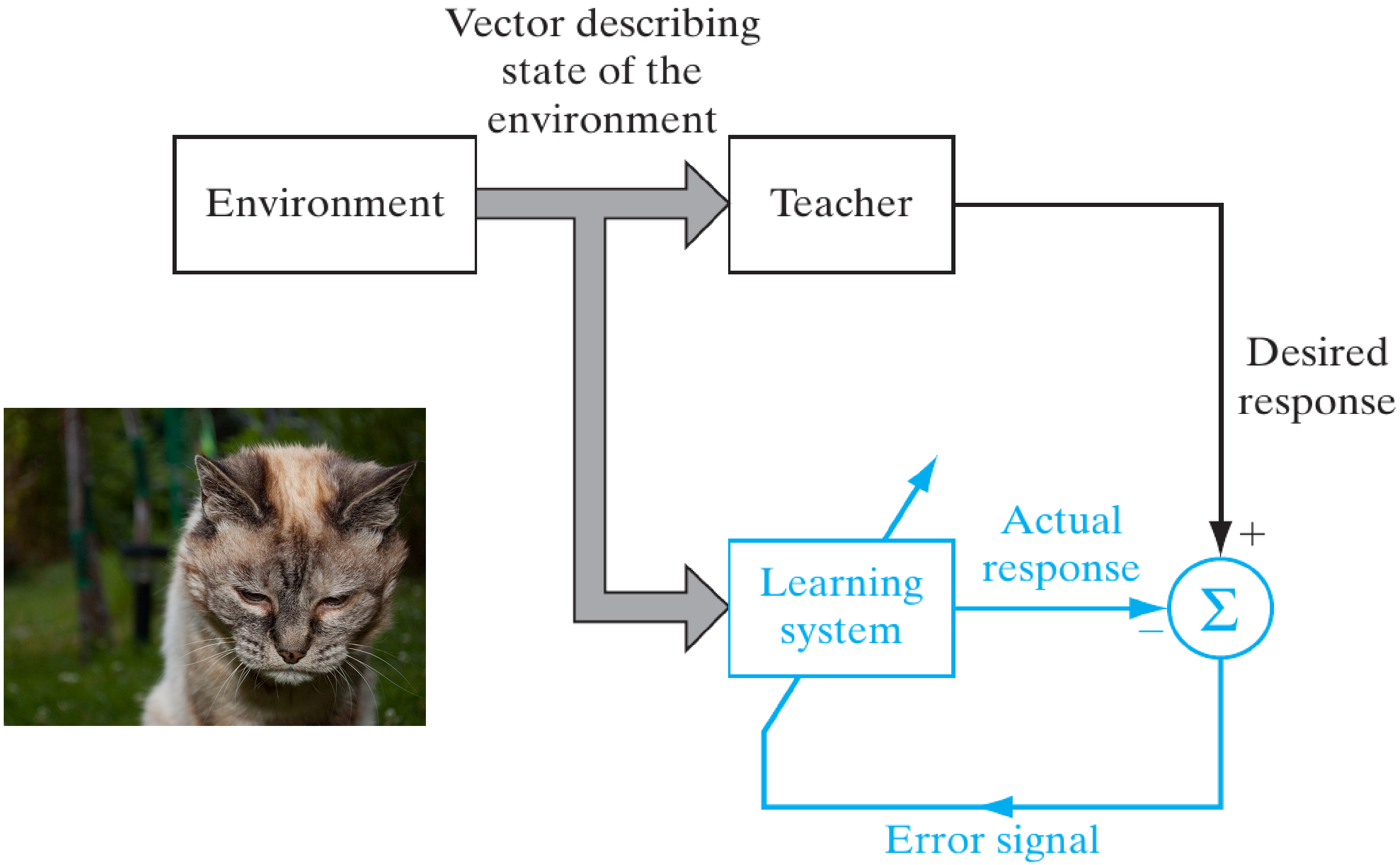
- Machine learning and artificial neural networks
  - Forms of learning and knowledge representation
  - Fully connected and convolutional neural networks, activation functions
  - Loss function, backpropagation
  - Overfitting, generalization
  - Machine learning surrogate models and physics-informed representation
- Example: Plasma-surface model interface
  - Fully-connected artificial neural network
  - Dimensionality reduction for extended TRIDYN data
  - Regression using variational autoencoder and mapper
- Concluding remarks

Machine learning and artificial neural networks

Fundamental aspects

Forms of learning: Supervised

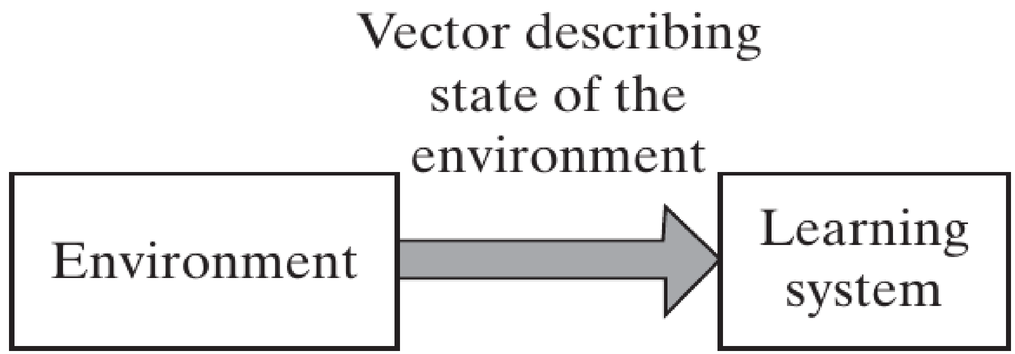
Teacher labels **desired response**



Haykin (2007) Neural networks and learning machines, Pearson Education

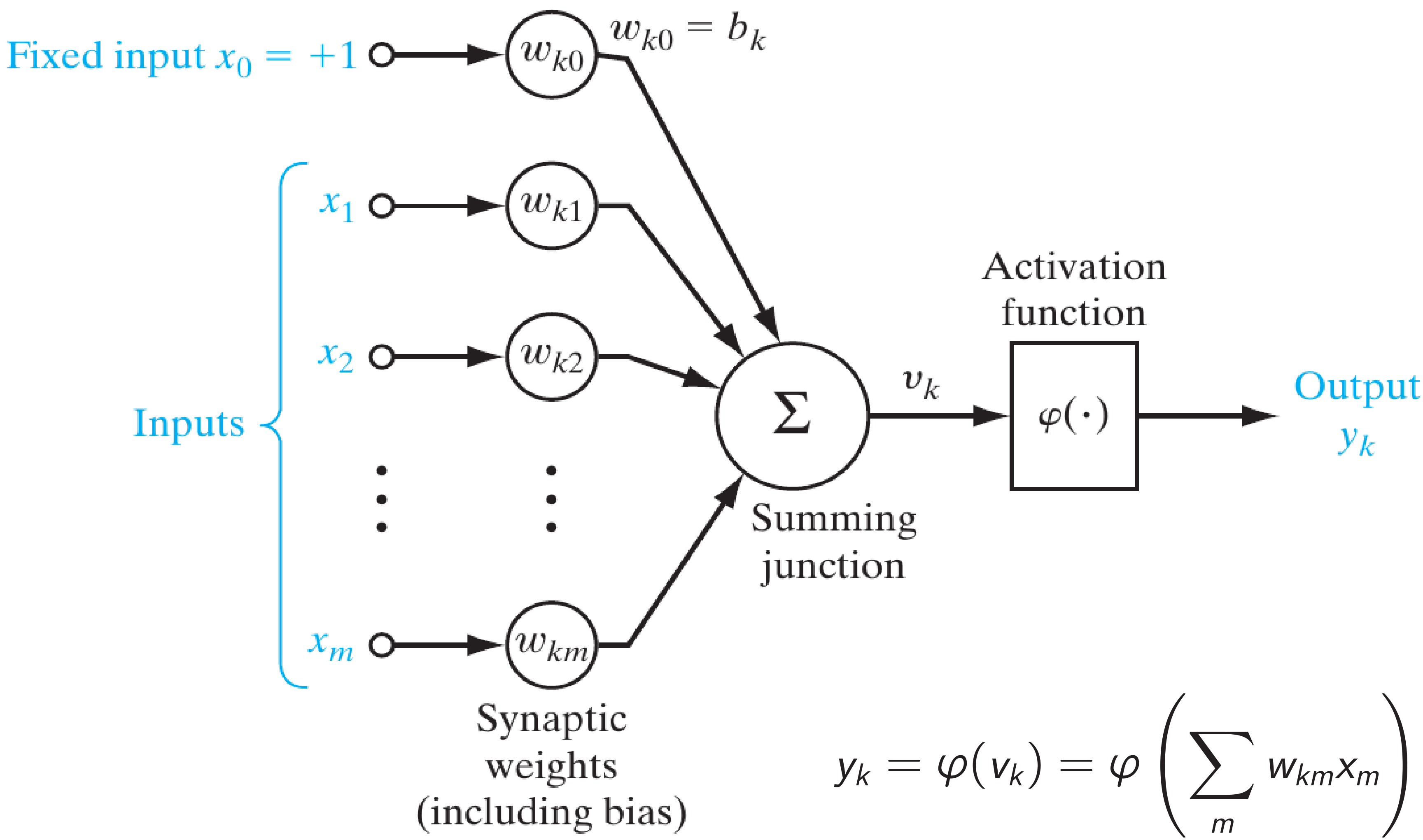
Forms of learning: Unsupervised

No teacher or critic, but **task independent measure** for quality of representation



Haykin (2007) Neural networks and learning machines, Pearson Education
Eatcha / CC BY-SA (https://creativecommons.org/licenses/by-sa/4.0)
Leuchtender Hund / CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0)
Marco Singer (Caronna) / CC BY-SA (http://creativecommons.org/licenses/by-sa/3.0/)

Knowledge representation with artifical synapses and neurons

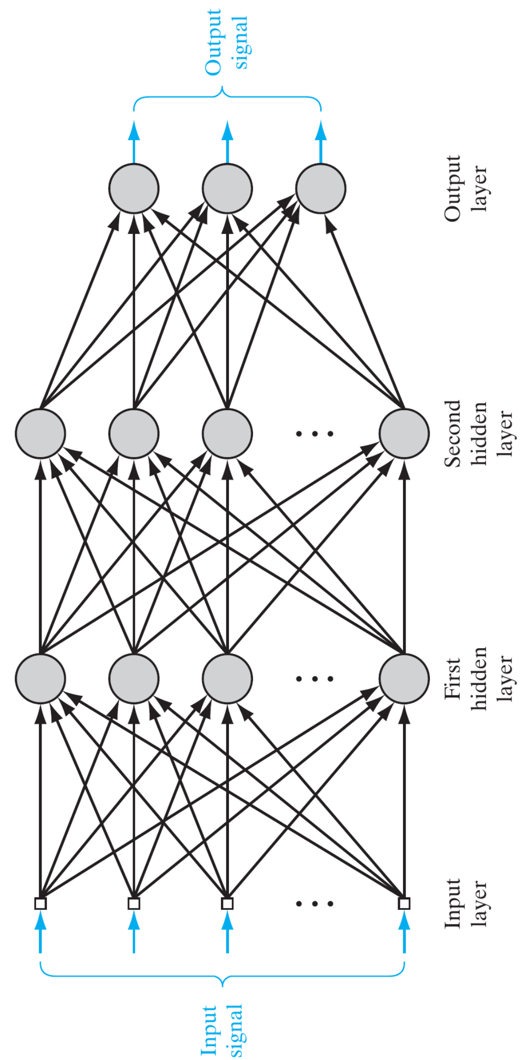


$$y_k = \varphi(v_k) = \varphi\left(\sum_m w_{km} x_m\right)$$

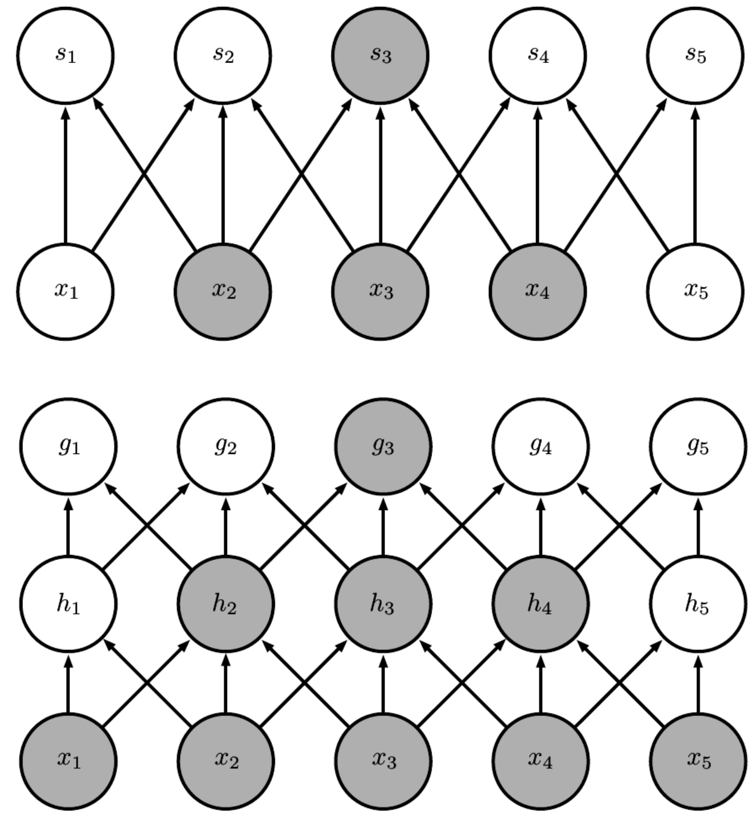
Haykin (2007) Neural networks and learning machines, Pearson Education

Knowledge representation with artifical neural networks

Fully connected network structure



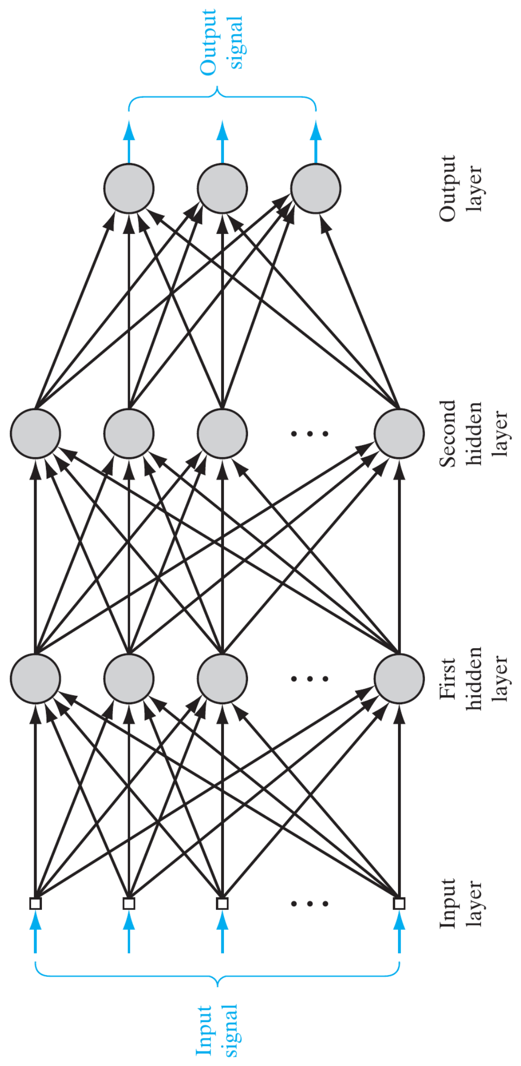
Convolutional network structure



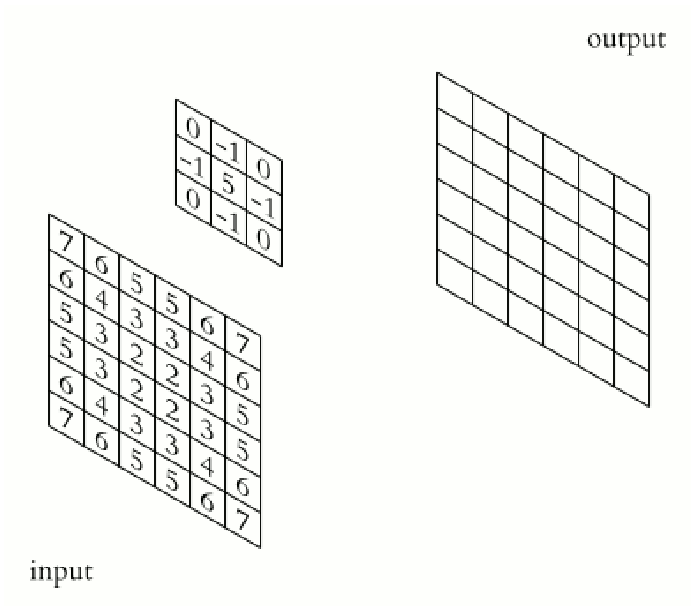
Haykin (2007) Neural networks and learning machines, Pearson Education
Goodfellow, Bengio, Courville (2016) Deep Learning, MIT Press

Knowledge representation with artifical neural networks

Fully connected network structure

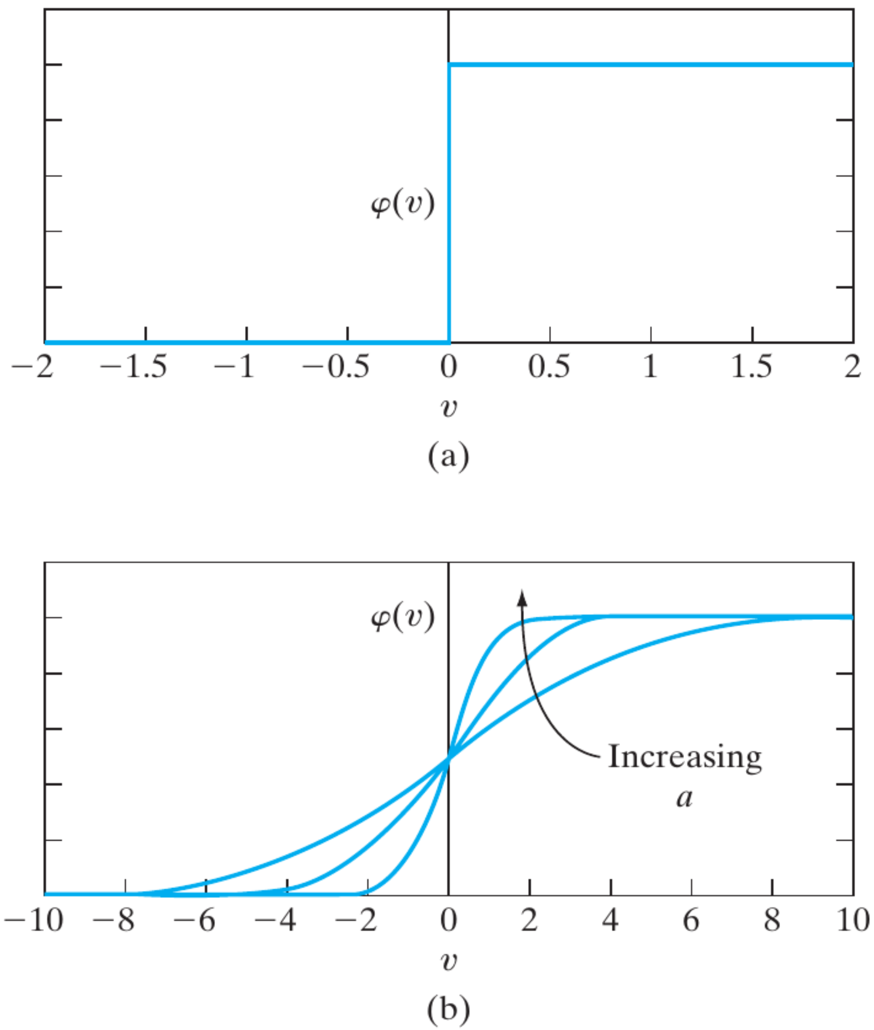


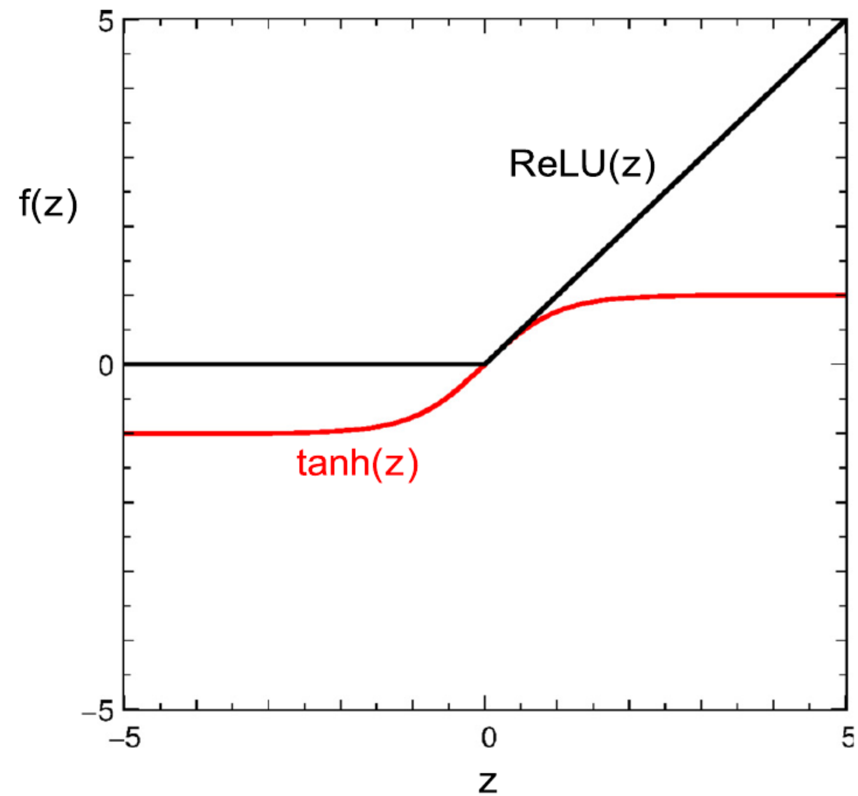
Convolutional network structure



Haykin (2007) Neural networks and learning machines, Pearson Education
`https://de.wikipedia.org/wiki/Datei:3D_Convolution_Animation.gif`

Activation functions





Threshold, sigmoid, hyperbolic tangent, rectified linear unit functions, etc

---

Haykin (2007) Neural networks and learning machines, Pearson Education
Spears et al., Phys. Plasmas 25, 080901 (2018)

Distance and loss function

Manhattan distance (L^1 norm)

$$D_1(\vec{p},\vec{q}) = ||\vec{p}-\vec{q}||_1 = \sum_{k=1}^{N} |p_k - q_k|$$

All components equally weighted

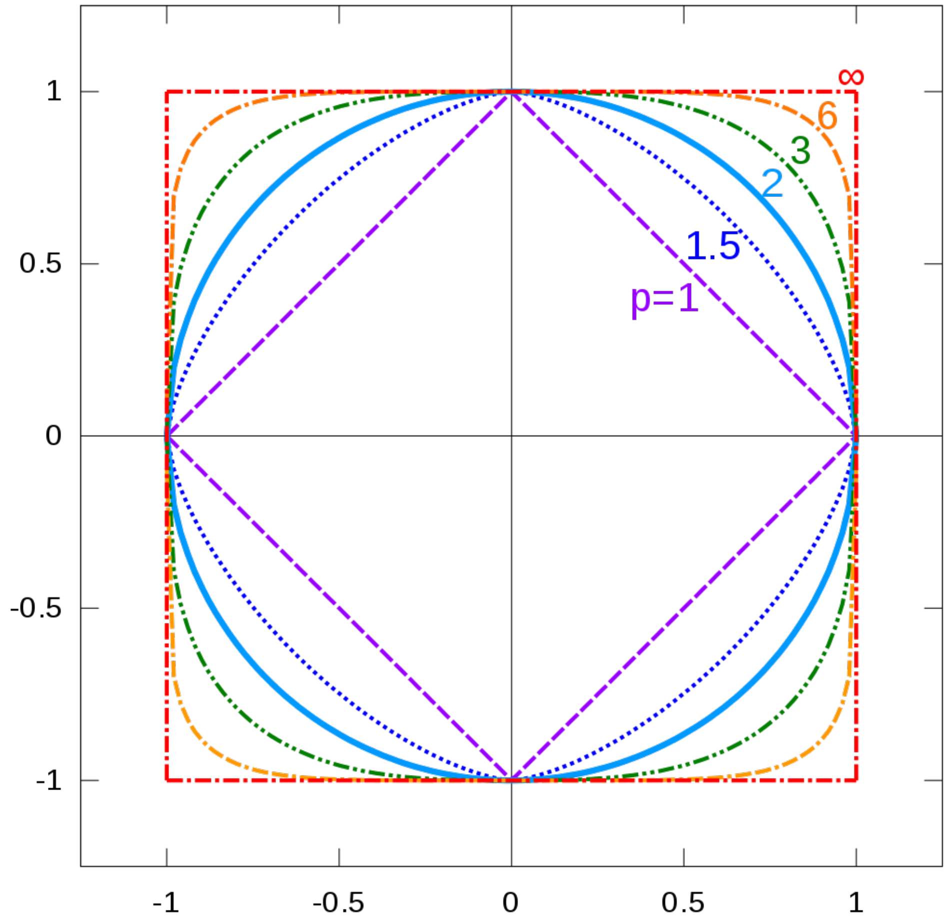
Euclidean distance (L^2 norm)

$$D_2(\vec{p},\vec{q}) = ||\vec{p}-\vec{q}||_2 = \sqrt{\sum_{k=1}^{N} |p_k - q_k|^2}$$

Gives more weight to large outliers

Loss / cost function for example

$$\mathcal{E}(\vec{w}) = \frac{1}{2}\sum_{\vec{x}_i\in X} D_p^2(\vec{y}_i^{\,*}(\vec{x}_i,\vec{w}),\vec{y}_i(\vec{x}_i))$$



https://commons.wikimedia.org/wiki/File:Vector-p-Norms_qtl1.svg

Backpropagation

Error signal of output neuron j:

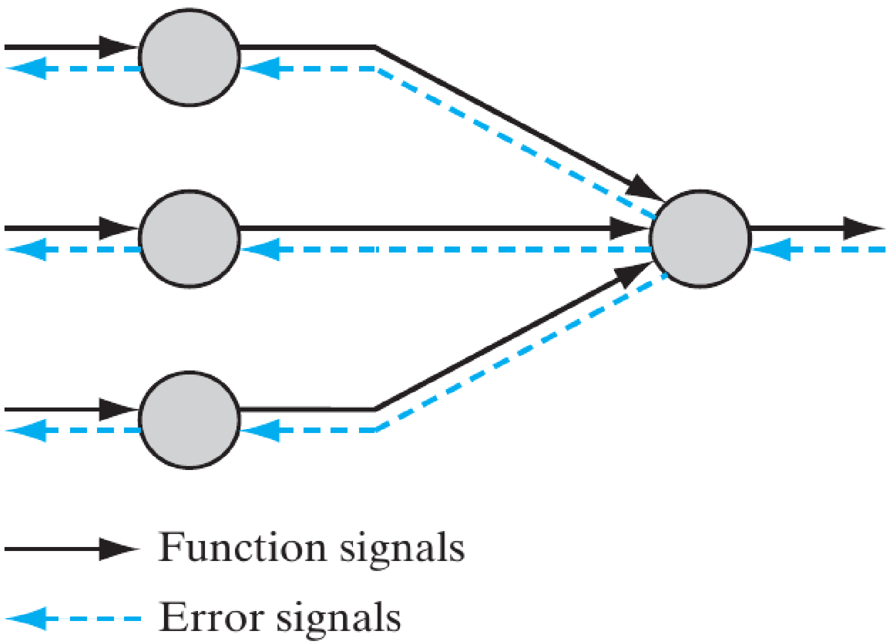
$$e_j(n) = d_j(n) - y_j(n)$$

Total instantaneous error energy:

$$\mathcal{E}(n) = \frac{1}{2}\sum_j e_j^2(n)$$

Output layer weight correction:

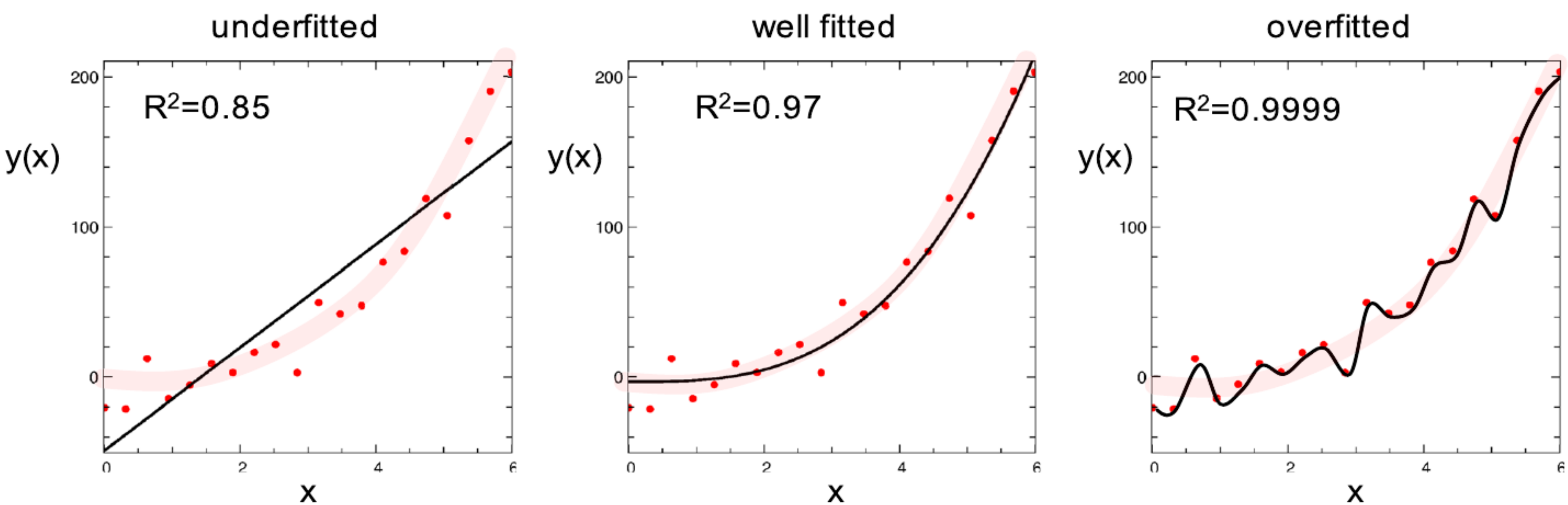
$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$



Haykin (2007) Neural networks and learning machines, Pearson Education

Overfitting and generalization

False adoption to uncertainties in training data due to available network complexity



Spears et al., Phys. Plasmas 25, 080901 (2018)

Physics-informed machine learning

Physics-informed machine learning

Set of partial differential equations

$$\mathcal{L}u(x) = f, \qquad x \in \Omega$$

$$\mathcal{B}u(x) = g, \qquad x \in \partial\Omega$$

Objective function

$$\mathcal{E} = \int_{\Omega} ||\mathcal{L}u^* - f||^2 dV + \int_{\partial\Omega} ||\mathcal{B}u^* - g||^2 dS$$

Optimization problem: Find $u^*(x, w_i)$ that minimizes $\mathcal{E}$

Dissanayake and Phan-Thien, "Neural-network-based approximations for solving partial differential equations", Comm. Num. Methods Eng. 10, 195 (1994)

Physics-informed deep learning

Physics-informed neural networks (PINNs) using automatic differentiation [1]

- Data-driven solutions of partial differential equations
- Data-driven discovery of partial differential equations

Low-temperature plasma modeling applications

- Predictive, Data-Driven Model for the Anomalous Electron Collision Frequency in a Hall Effect Thruster [2]
- Machine Learning Plasma-Surface Interface for Coupling Sputtering and Gas-Phase Transport Simulations [3]
- Fast prediction of electron-impact ionization cross sections of large molecules via machine learning [4]
- Deep learning for solving the Boltzmann equation of electrons in weakly ionized plasma [5]
- Determining cross sections from transport coefficients using deep neural networks [6]
- Deep learning for thermal plasma simulation: Solving 1-D arc model as an example [7]

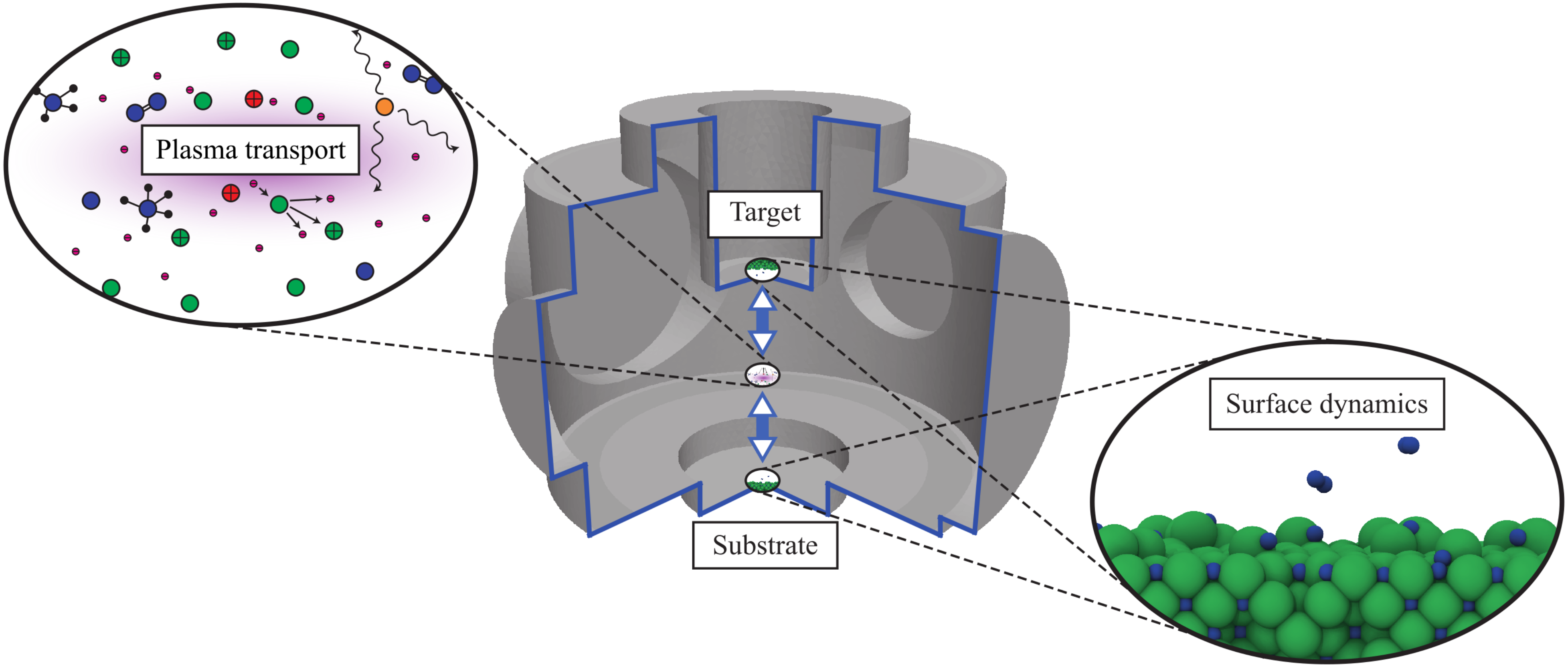
---

[1] Raissi, Perdikaris, Karniadakis, J. Comp. Phys. 378, 686 (2019)
[2] Jorns, Plasma Sources Sci. Technol. 27, 104007 (2018)
[3] Krüger, Gergs, Trieschmann, Plasma Sources Sci. Technol. 28, 035002 (2019)
[4] Zhong, Journal of Applied Physics 125, 183302 (2019)
[5] Kawaguchi, Takahashi, Ohkama, Satoh, Plasma Sources Sci. Technol. 29, 025021 (2020)
[6] Stokes, Cocks, Brunger, White, Plasma Sources Sci. Technol. 29, 055009 (2020)
[7] Zhong, Gu, Wu, Computer Physics Communications 257, 107496 (2020)

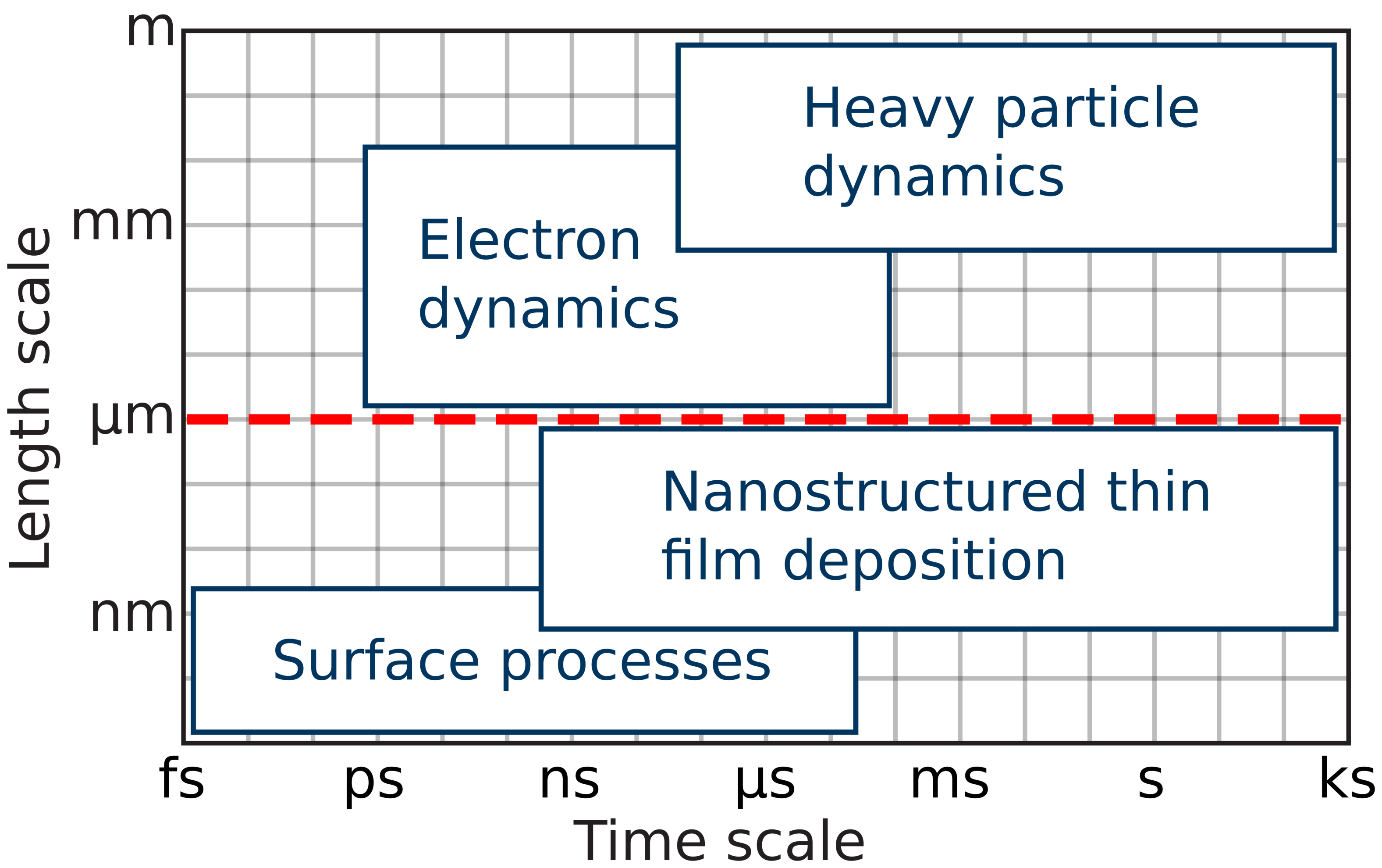
Example: Plasma-surface model interface

F. Krüger, T. Gergs, and J. Trieschmann, Plasma Sources Science and Technology 28, 035002 (2019)
DOI: 10.1088/1361-6595/ab0246

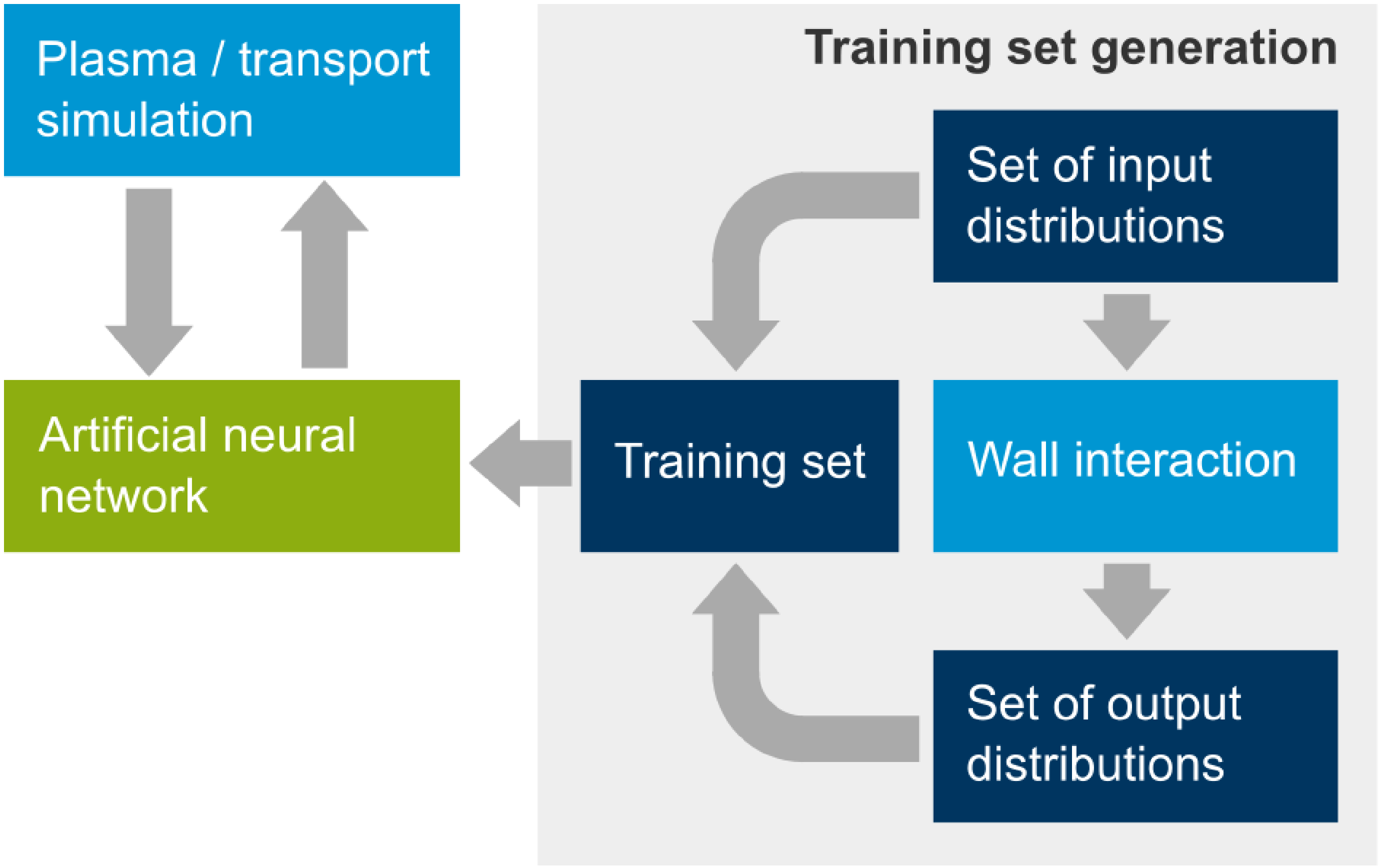
Low-temperature model representation



Physical scales



Machine learning plasma-surface interface

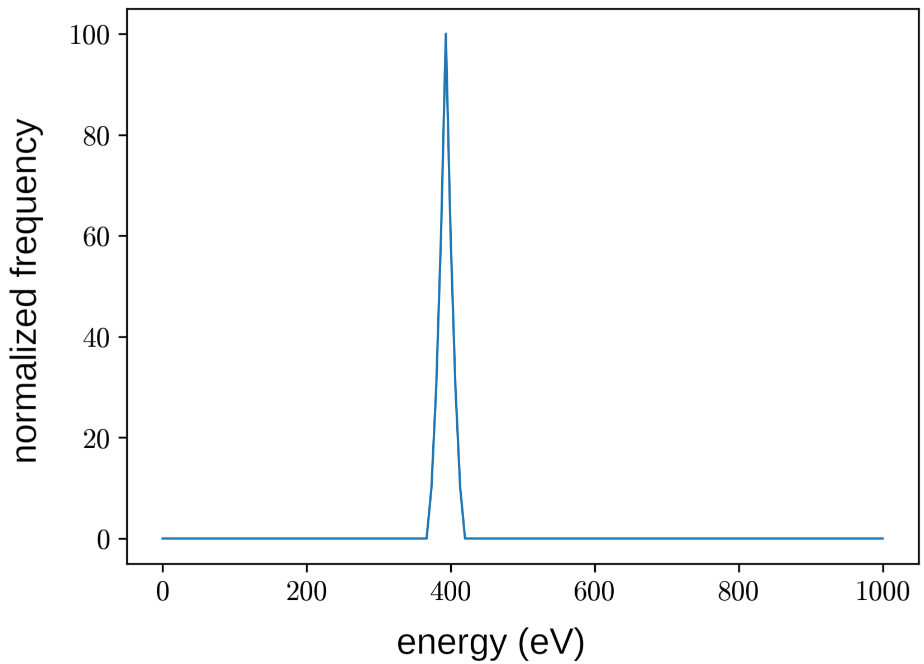


2019: Fully-connected artificial neural network

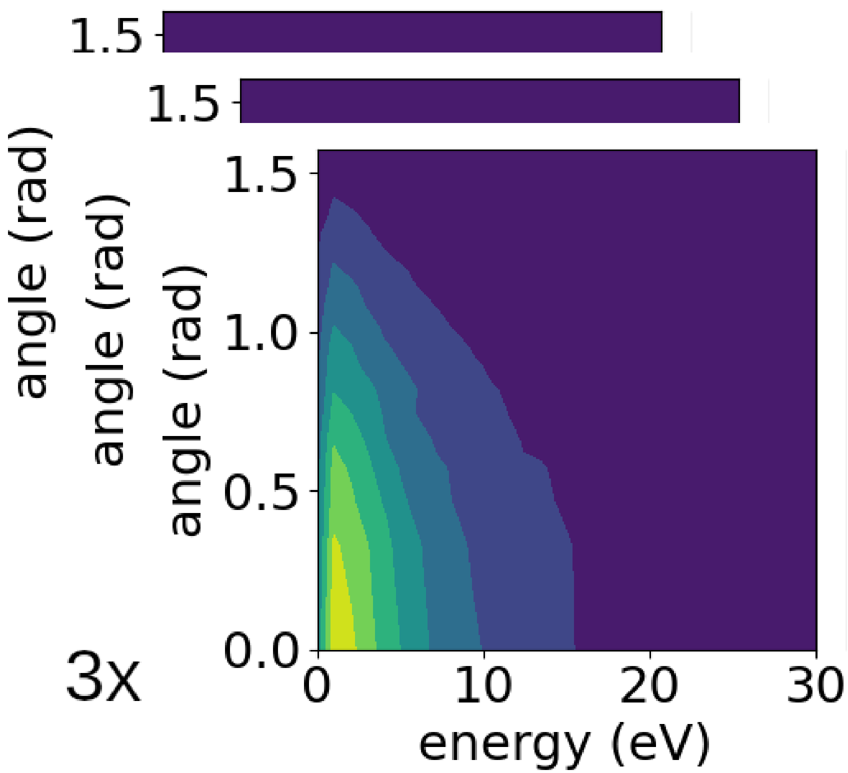
F. Krüger, T. Gergs, and J. Trieschmann, Plasma Sources Science and Technology 28, 035002 (2019)
DOI: 10.1088/1361-6595/ab0246

Data set: Ar on Ti(50%)-Al(50%) composite using TRIDYN

Input



Output

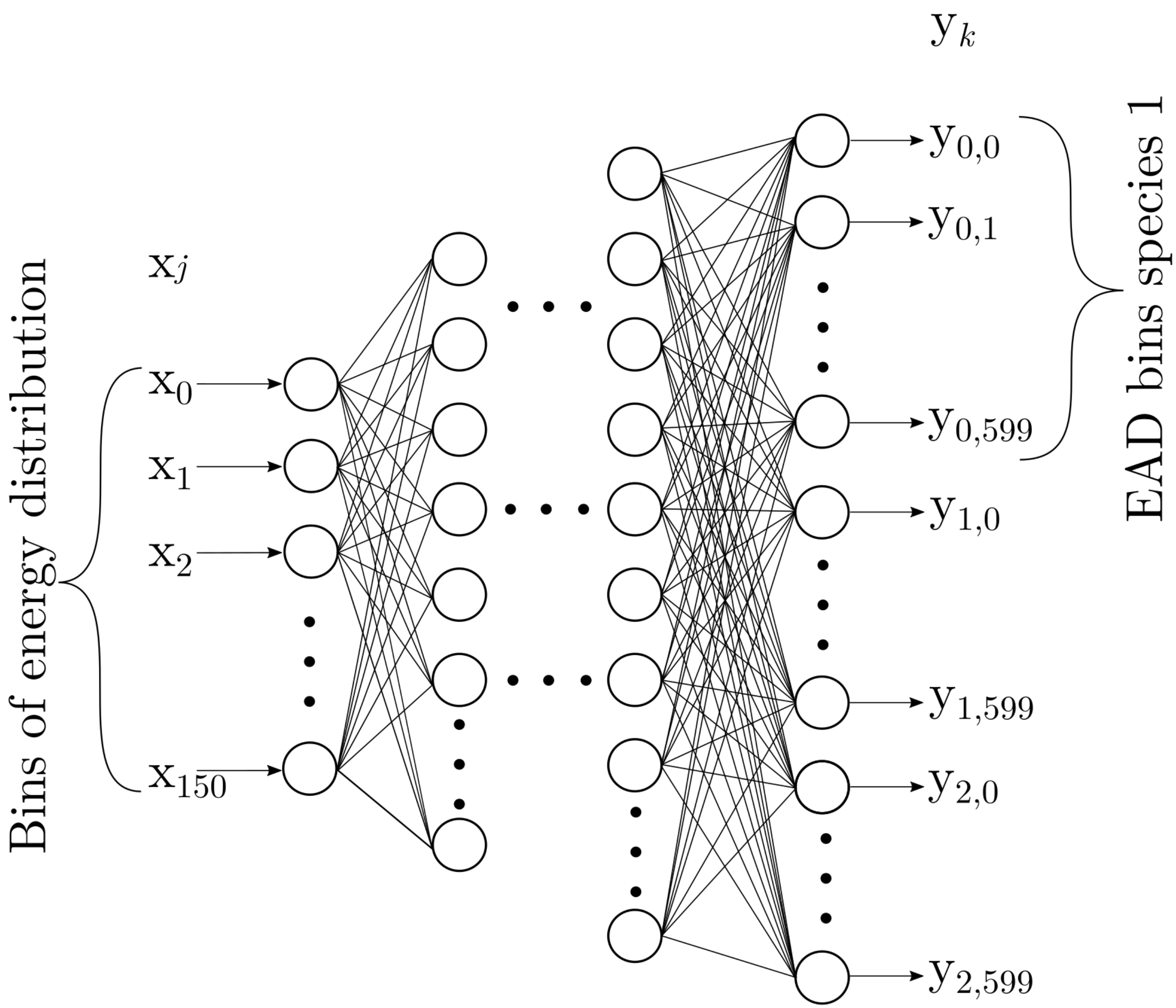


- 1D energy distribution histogram
- 439 combinations define total set

- 2D energy/angle histogram
- 3 histograms (1 for each species)

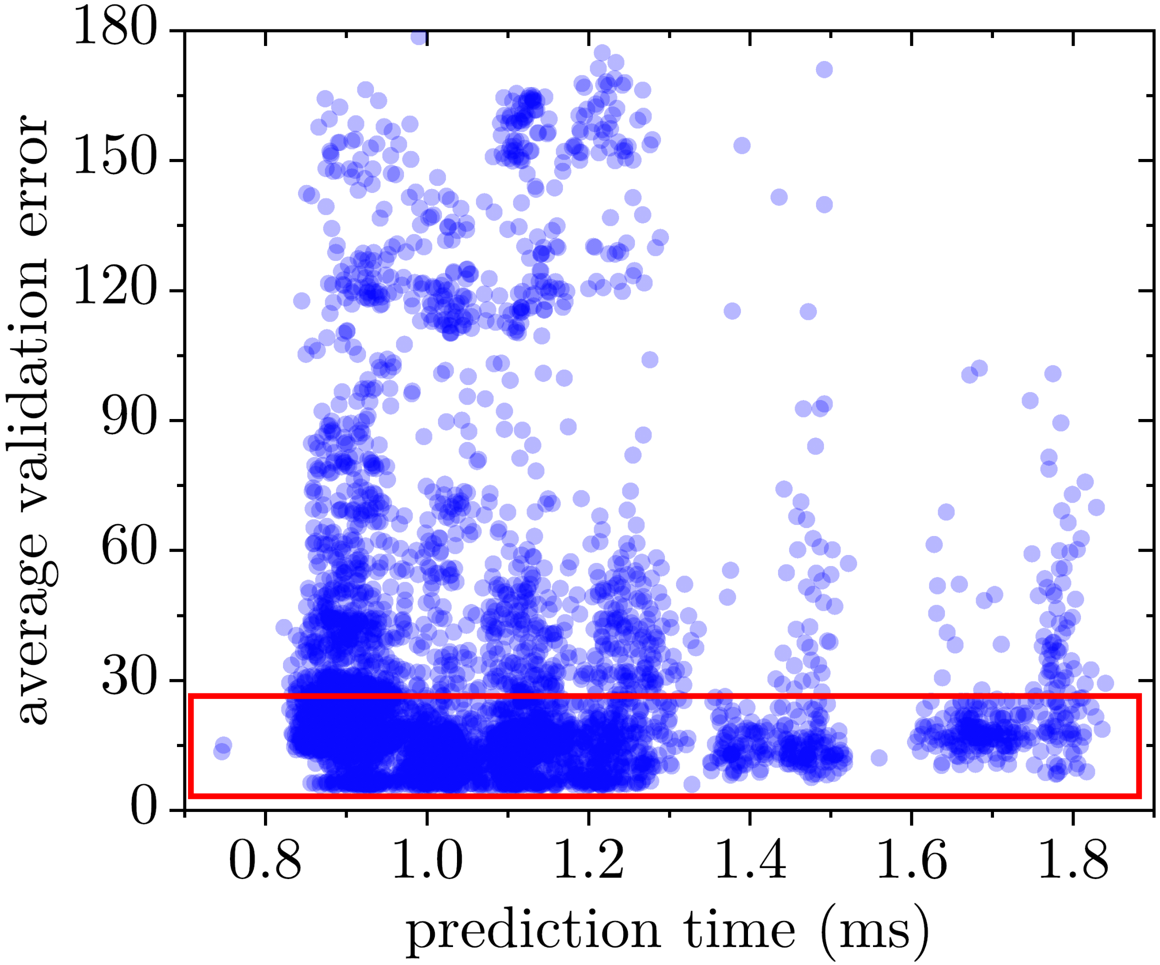
TRIDYN: W. Möller, W. Eckstein, Nucl. Instr. and Meth. Phys. Research Sect. B 2, 814 (1984)

Fully-connected artificial neural network



Artificial neural network simulated using Keras and TensorFlow, `www.tensorflow.org`

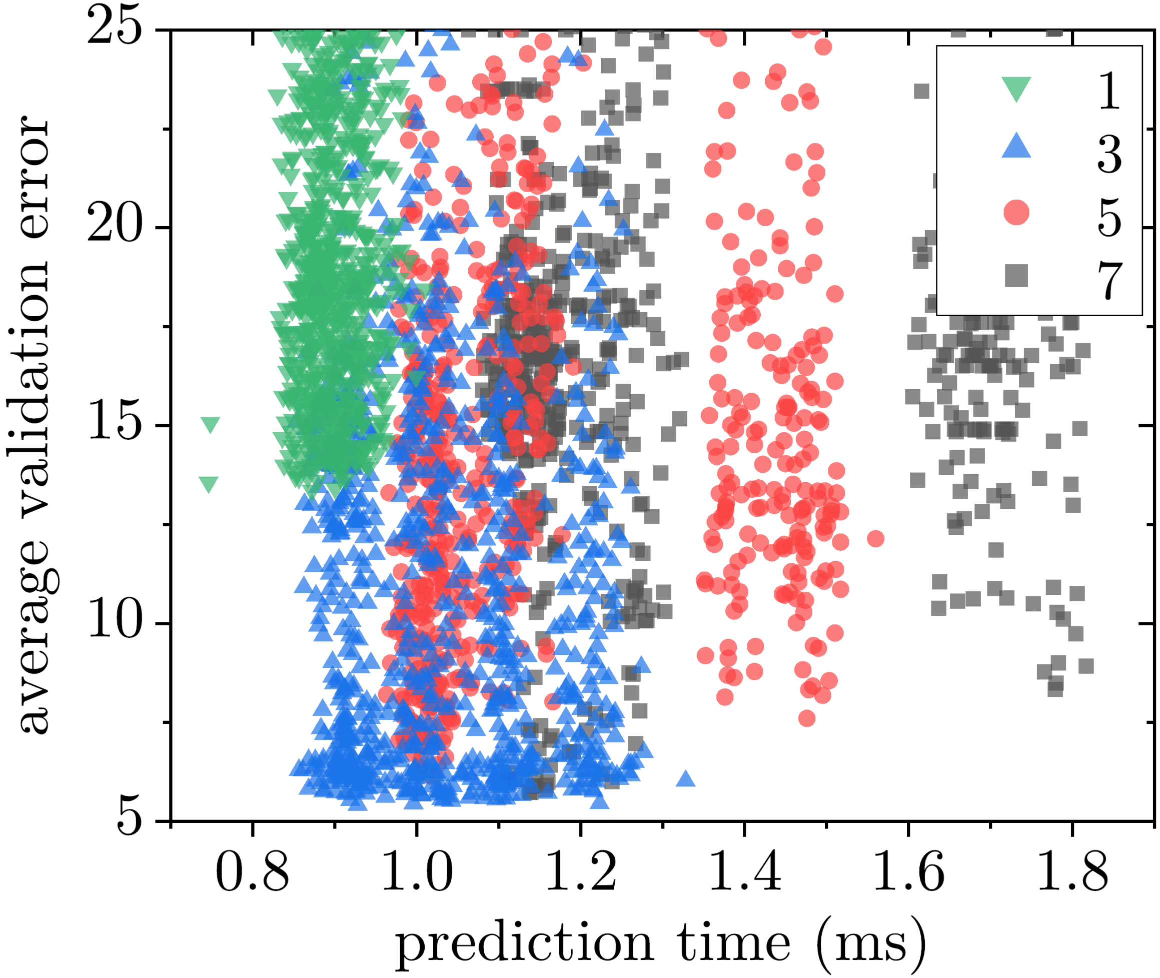
Hyperparameter space



(a)

- About 600 hyperparameter combinations
- All networks fulfill prediction time requirement

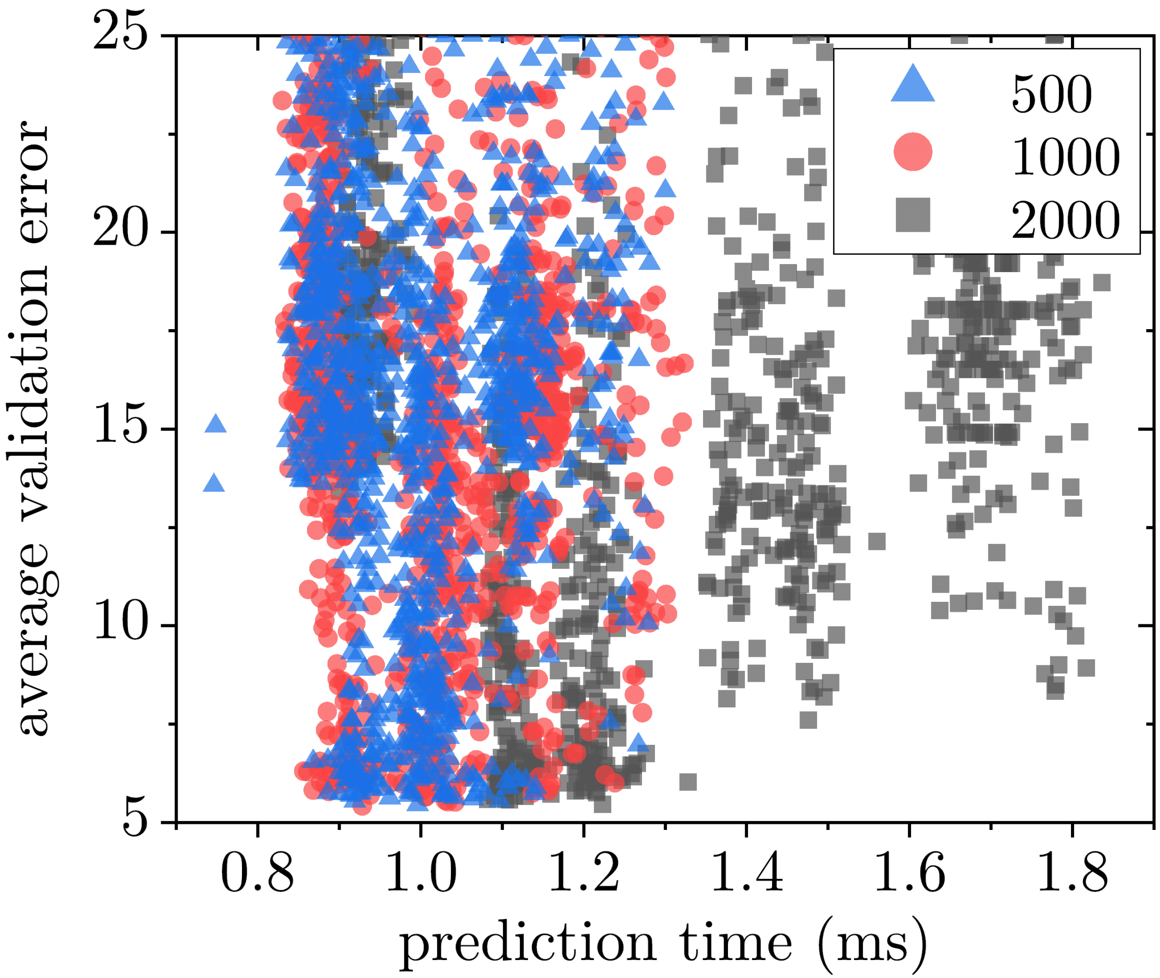
Hyperparameter space



(b)

- Variation: number of hidden layers
- All **multilayer networks** achieve low error

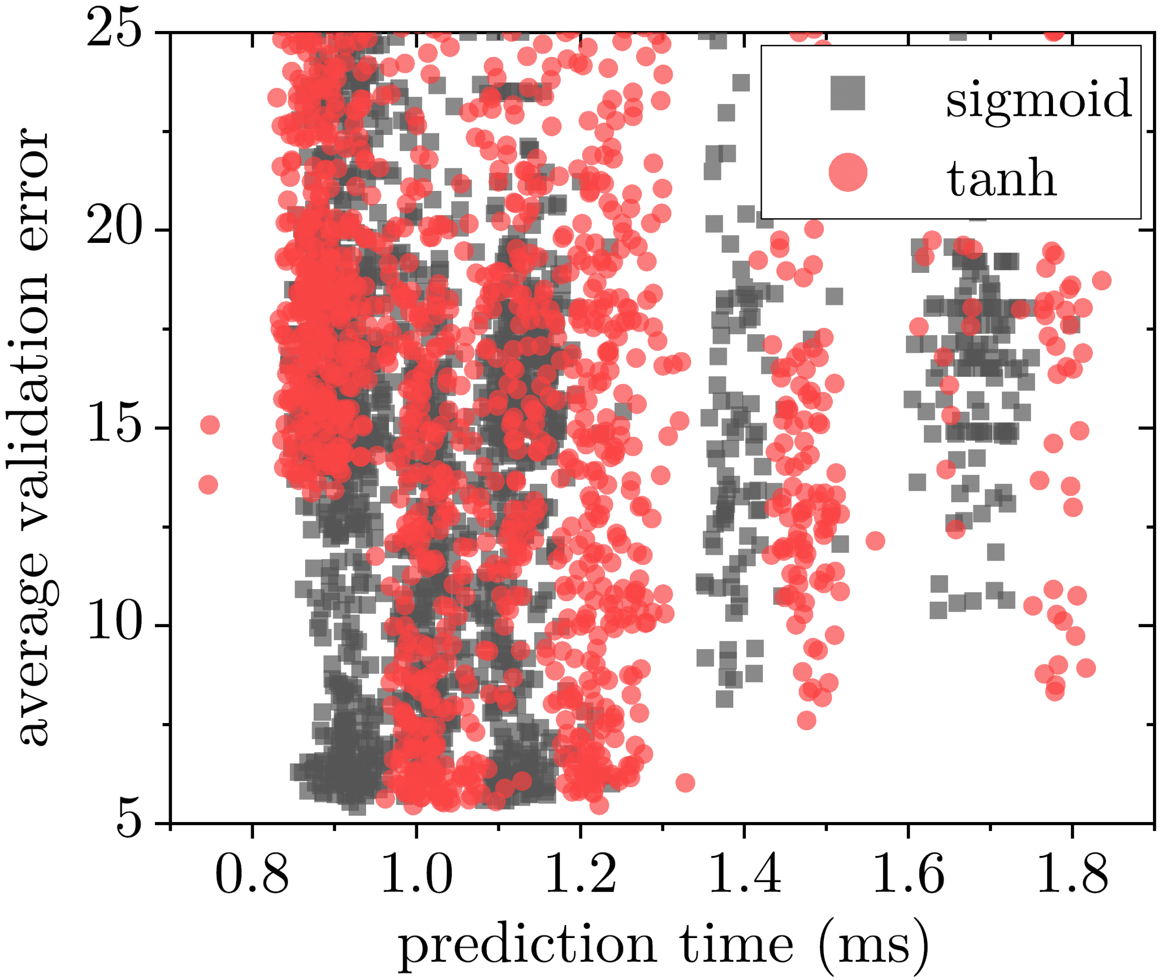
Hyperparameter space



(c)

- Variation: number of nodes per activation layer
- Complexity increases prediction time, **not necessarily quality**

Hyperparameter space



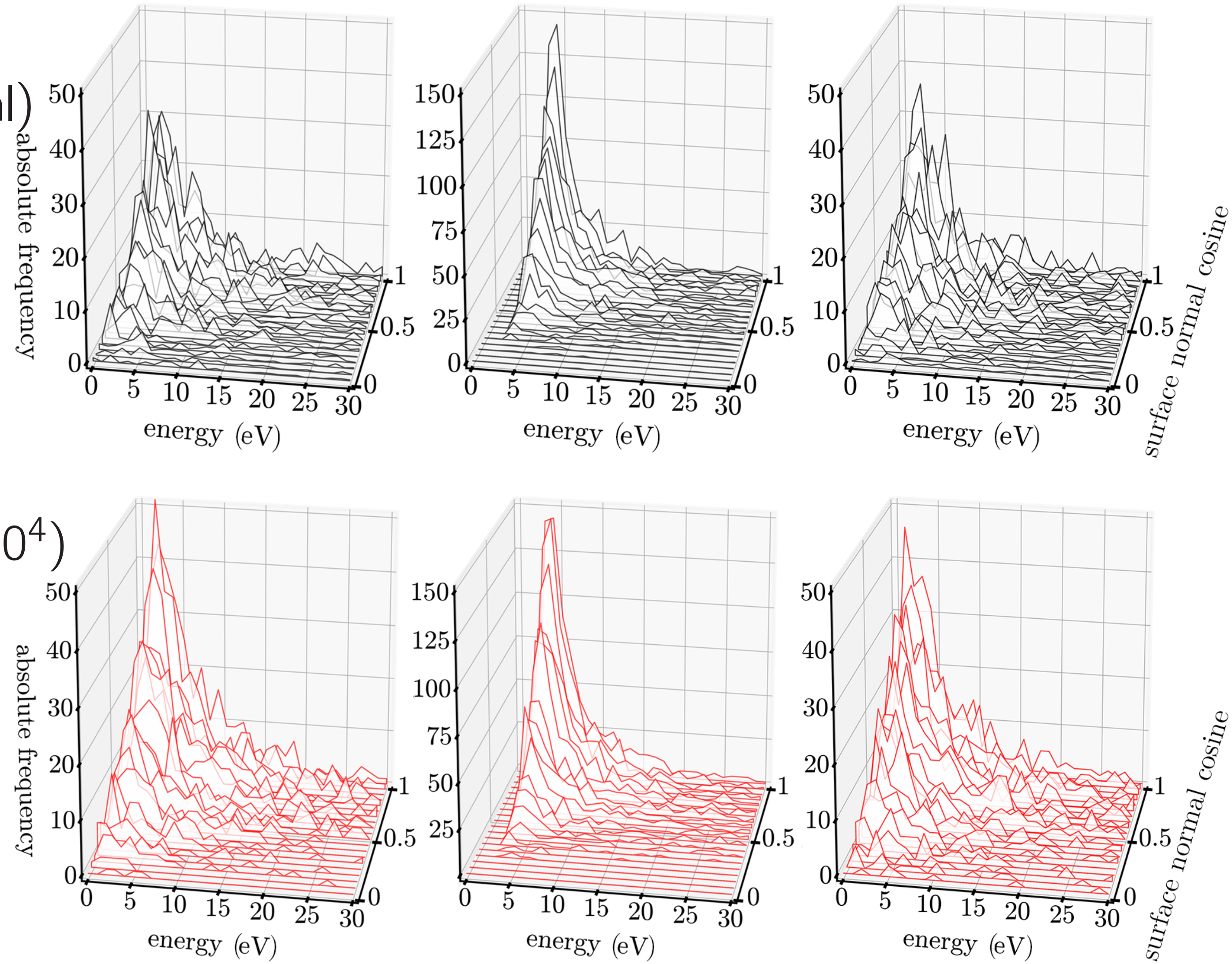
(d)

- Variation: activation function
- sigmoid less computational effort than tanh

Sub-optimal prediction (unknown data, 390 eV)

ANN (sub-optimal)

TRIDYN (10^4)



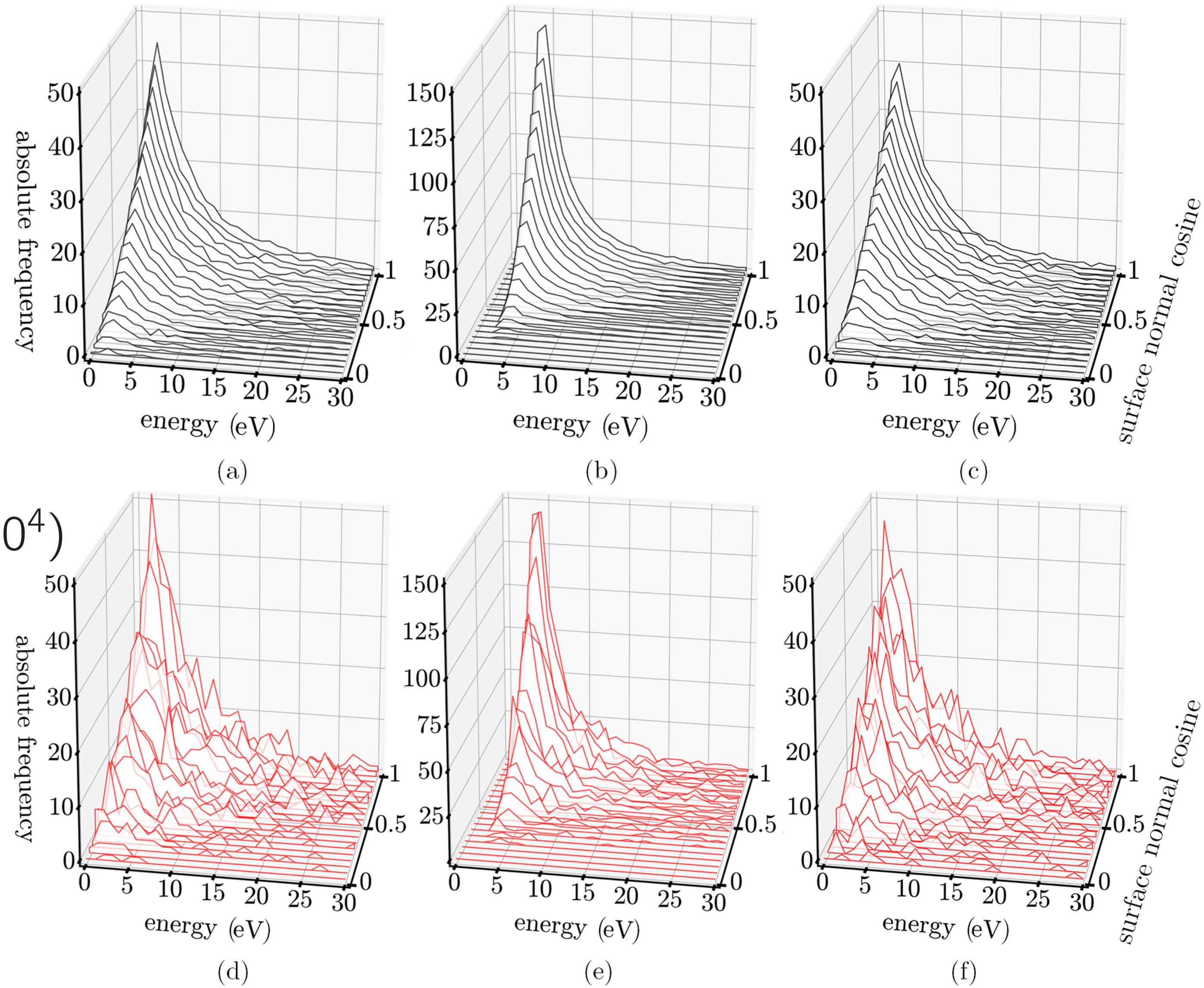
(d) (e) (f)

Generalization and statistical mitigation, $R^2_{\text{train}} = 0.961$ and $R^2_{\text{pred}} = 0.998$

Prediction (unknown data, 390 eV)

ANN

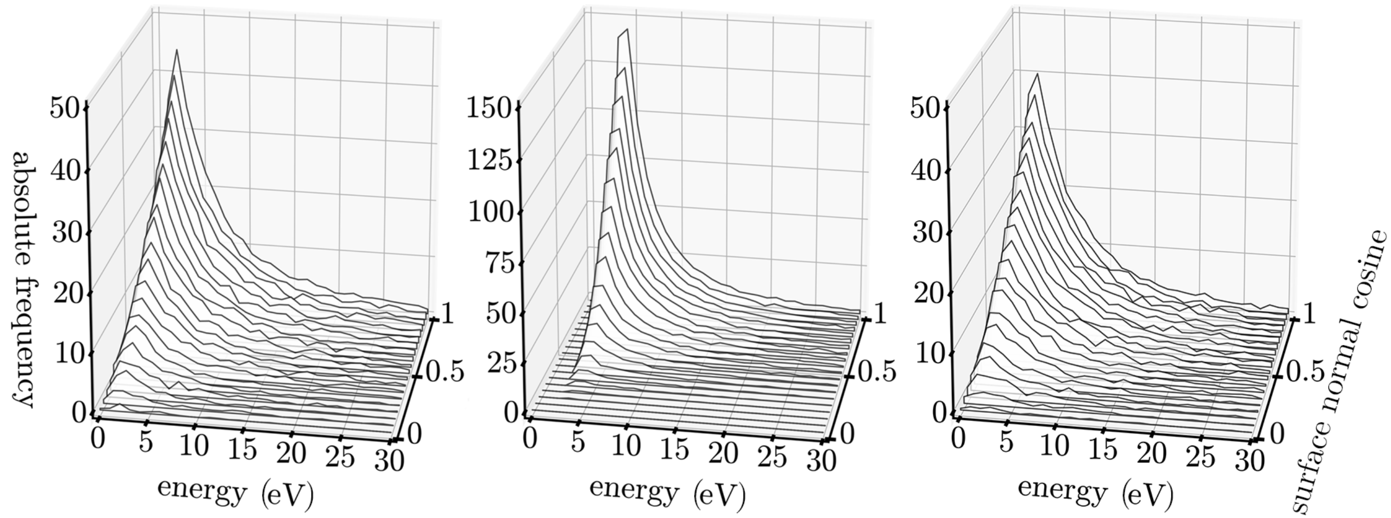
TRIDYN (10^4)



Generalization and statistical mitigation, $R^2_{\text{train}} = 0.961$ and $R^2_{\text{pred}} = 0.998$

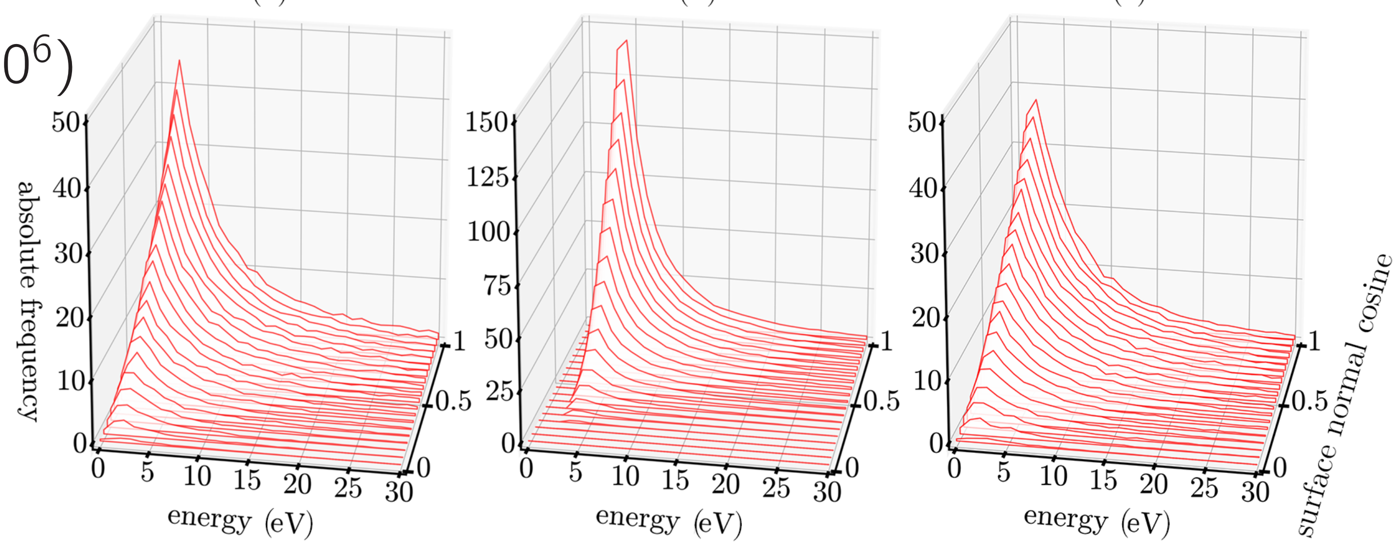
Prediction (unknown data, 390 eV)

ANN



(a) (b) (c)

TRIDYN (10^6)



Generalization and statistical mitigation, $R^2_{\text{train}} = 0.961$ and $R^2_{\text{pred}} = 0.998$

How complex need this network to be?

Final optimized fully connected network structure:

- 1×151 input layer
- 3×1000 hidden layers + bias
- 1×1800 output layer

Total of 3,955,800 weight parameters

Is this level of complexity really necessary?

2020: Dimensionality reduction for extended TRIDYN data

T. Gergs, B. Borislavov, and J. Trieschmann, in preparation

How many independent parameter dimensions required?

Maxwell-Boltzmann energy distribution $\rightarrow$ 1 degree of freedom

$$f(E)dE = 2N\sqrt{\frac{E}{\pi}\left(\frac{1}{kT}\right)^3}\exp\left(-\frac{E}{kT}\right)dE$$

Sigmund-Thompson energy-angular distribution $\rightarrow$ 4 degrees of freedom

$$f(E,\theta)\,dE\,d^2\Omega = F_{\mathrm{D}}(E_0,\theta_0)\,\frac{\Gamma_m}{4\pi}\frac{1-m}{NC_m}\frac{E}{(E+U)^{3-2m}}\cos(\theta)\,dE\,d^2\Omega$$

Behrisch, Eckstein (2007) Sputtering by Particle Bombardment, Springer

Data set: Sputtering Ar on Ti-Al composite using TRIDYN

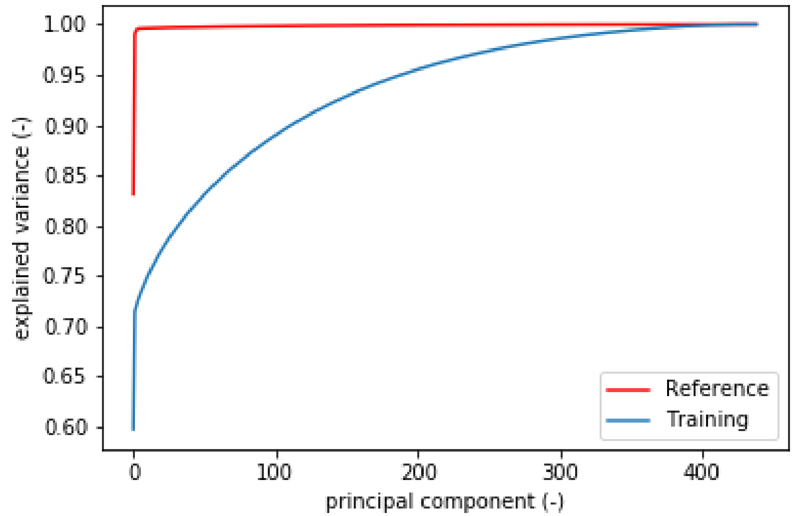
- Similar input/output relation as previous case
- 1D energy distribution histogram
- 439 combinations define single subset
- 2D energy/angle histogram
- 3 histograms (1 for each species)

- **3 different** chemical compositions $x = [0.3, 0.5, 0.7]$ (previously **single** stoichiometry $x = 0.5$)
- Total of 1317 cases with 80%, 10%, 10% train/validation/test split
- Complete set with 10^4 (training) and 10^6 (ground truth) projectiles statistics

---

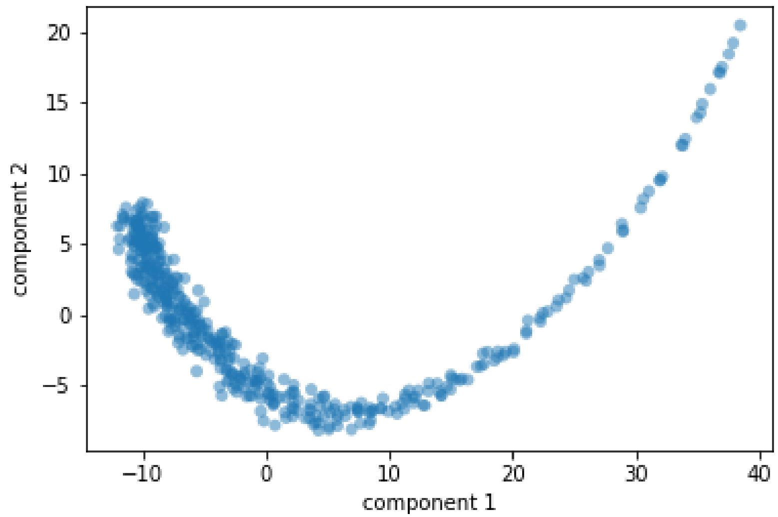
Input from TRIDYN: W. Möller, W. Eckstein, Nucl. Instr. and Meth. Phys. Research Sect. B 2, 814 (1984)

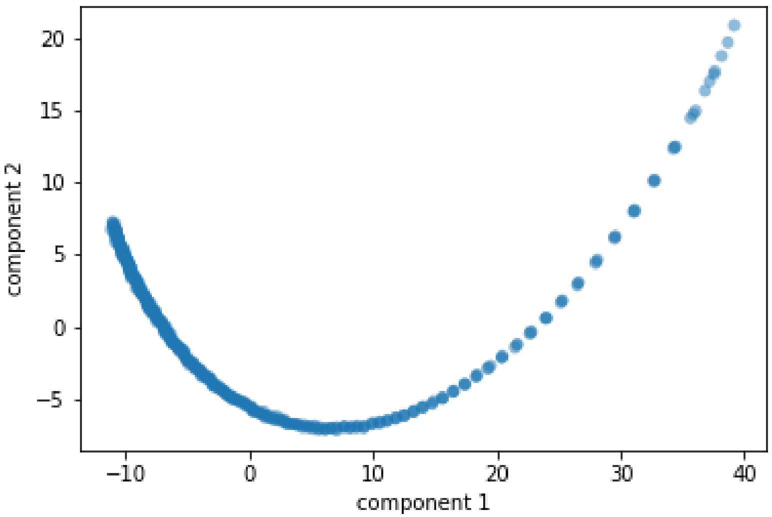
Principle component analysis (PCA)



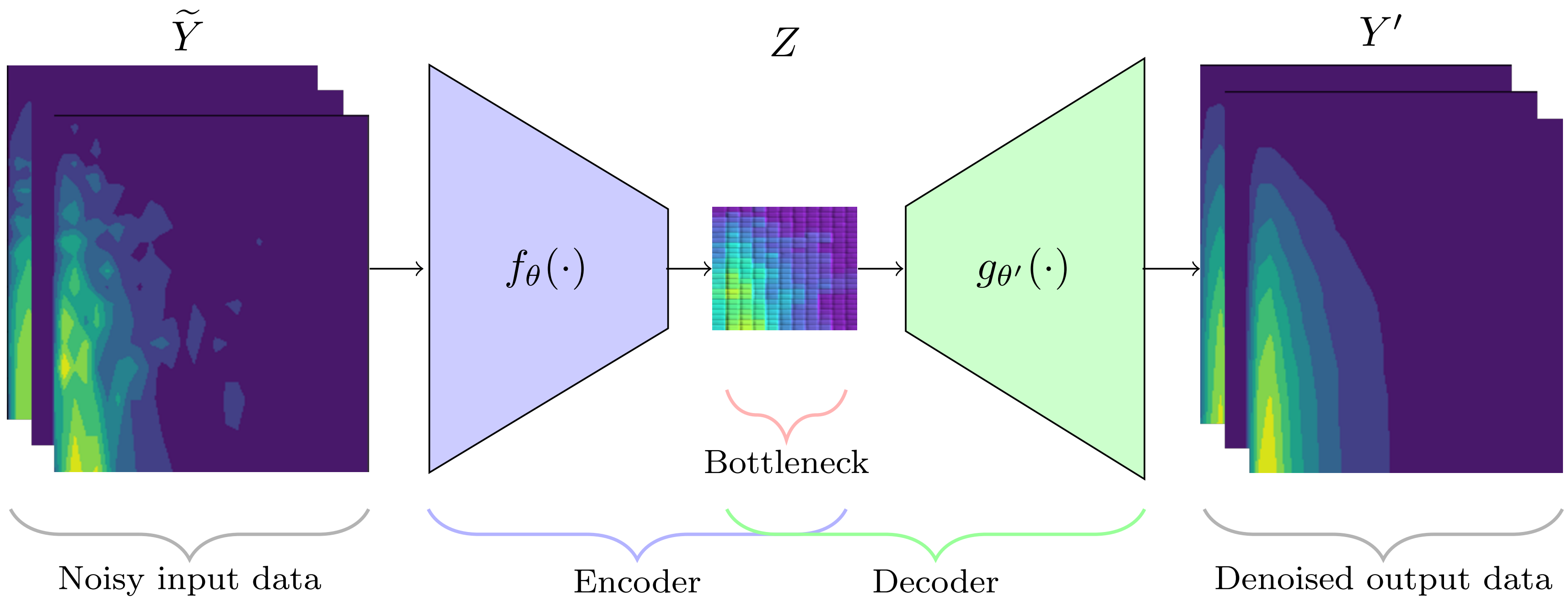
Training 10^4

Ground truth 10^6





Convolutional autoencoder network



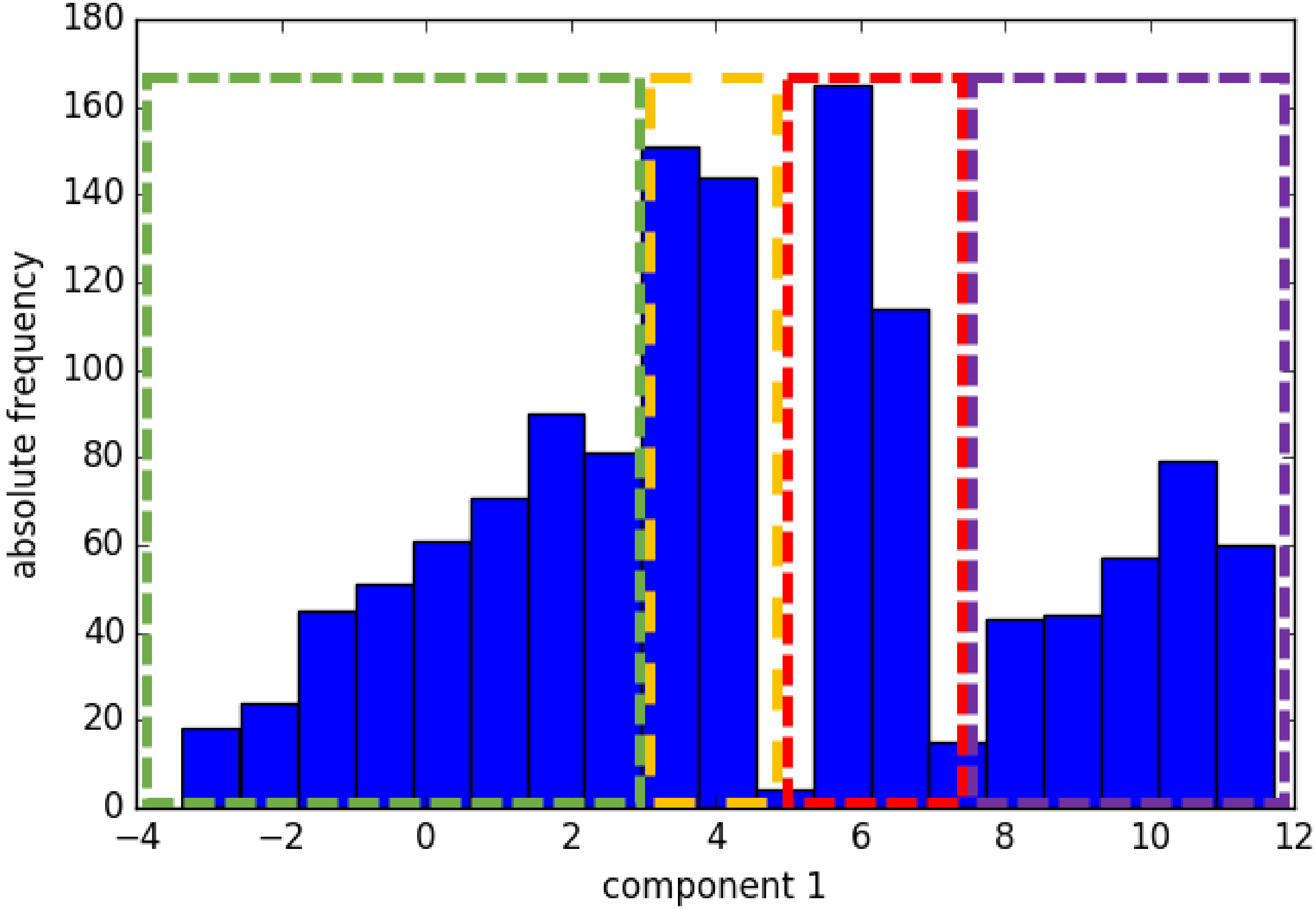
Bottleneck architecture with variable latent space dimensions

Dimensionality reduction to $D_c = 1 \dots 3$ latent space components

---

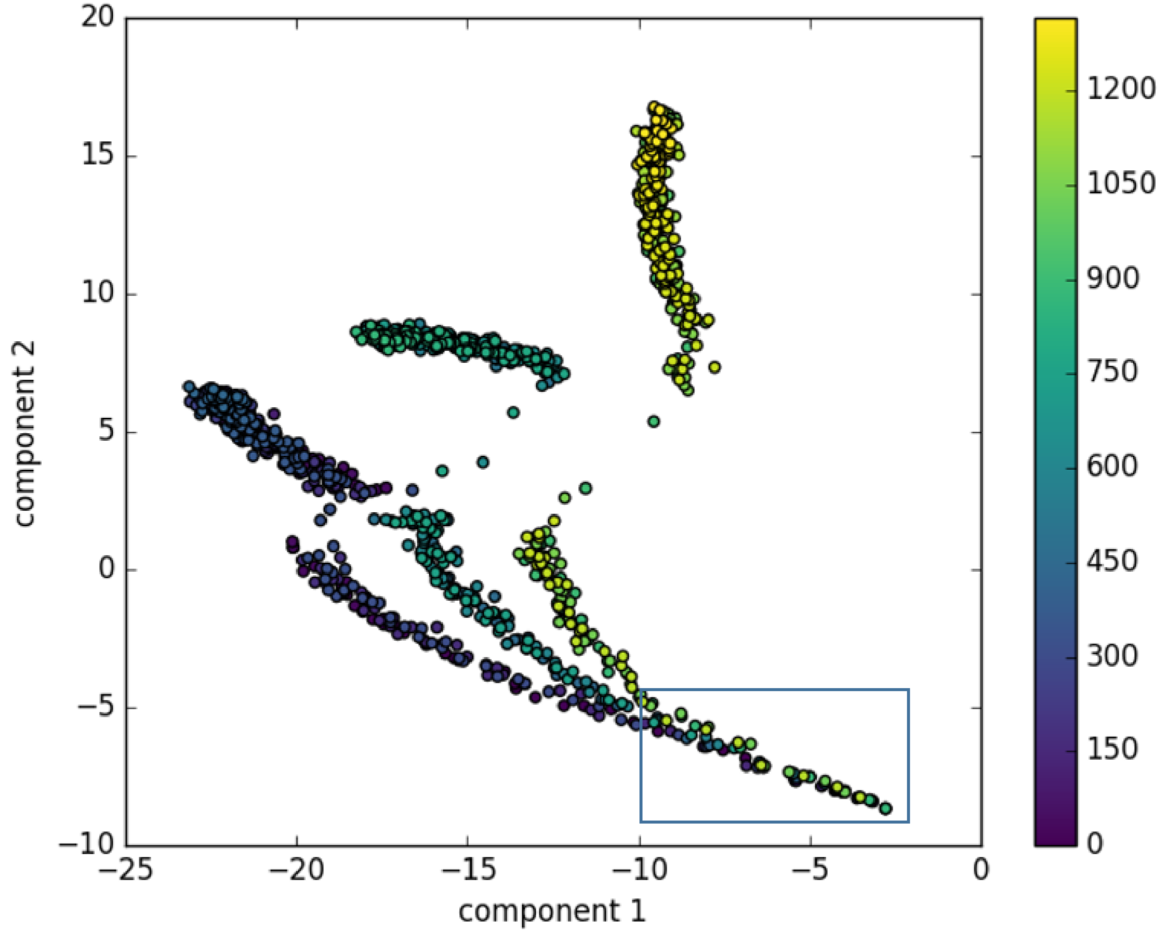
G.E. Hinton and R.R. Salakhutdinov, Science 313, 504 (2006)
Artificial neural network simulated using Keras and TensorFlow, `www.tensorflow.org`

Latent space representation



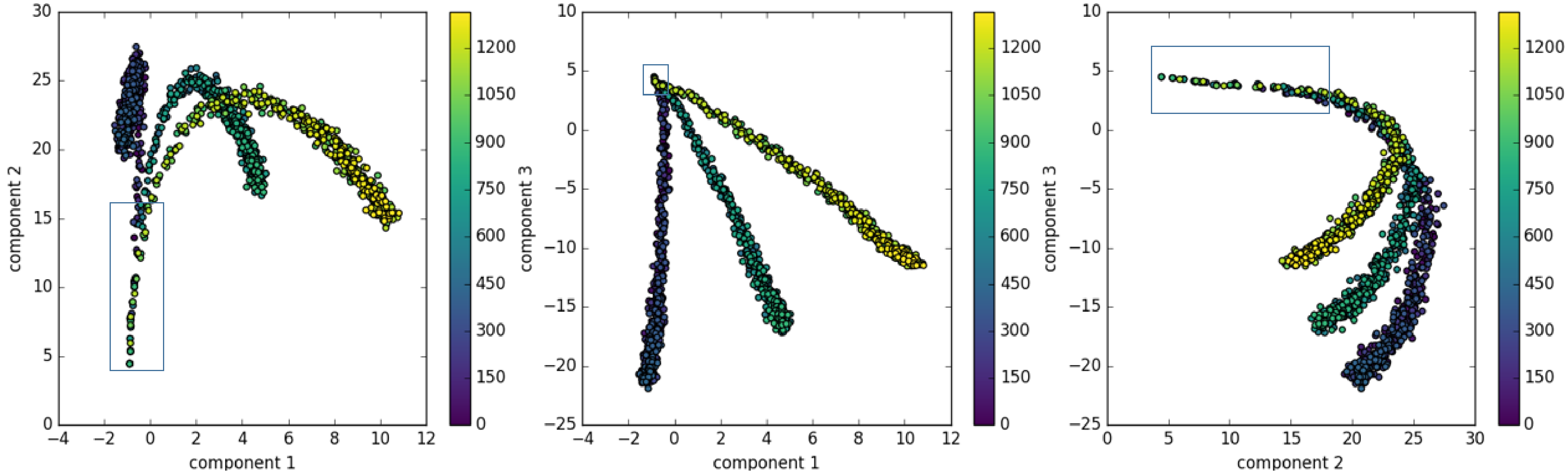
Bottleneck architecture with latent space dimensions $n = 1$

Latent space representation



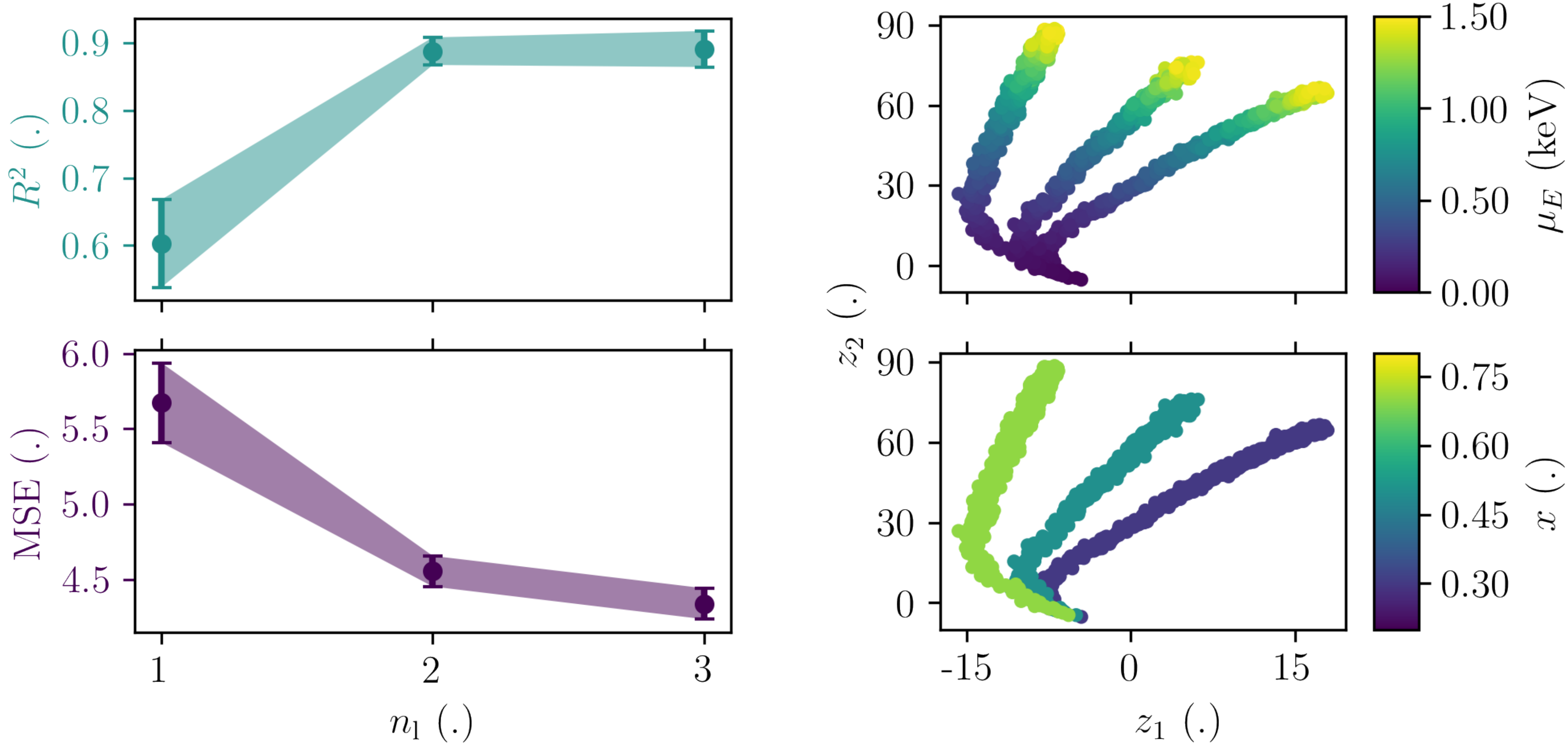
Bottleneck architecture with latent space dimensions $n = 2$

Latent space representation



Bottleneck architecture with latent space dimensions $n = 3$

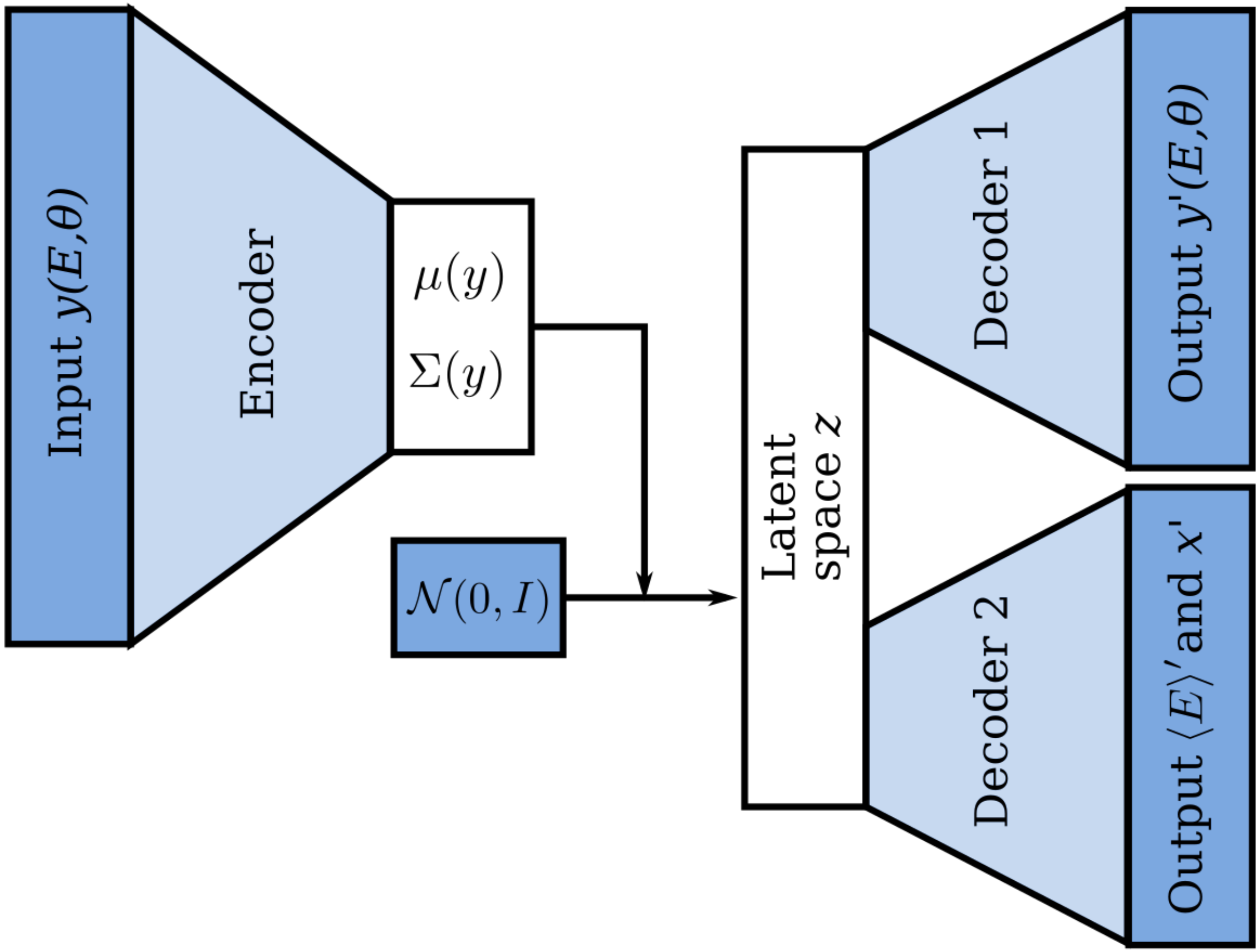
Latent space representation



2021: Regression using variational autoencoder and mapper

T. Gergs, B. Borislavov, and J. Trieschmann, in preparation

Variational autoencoder network

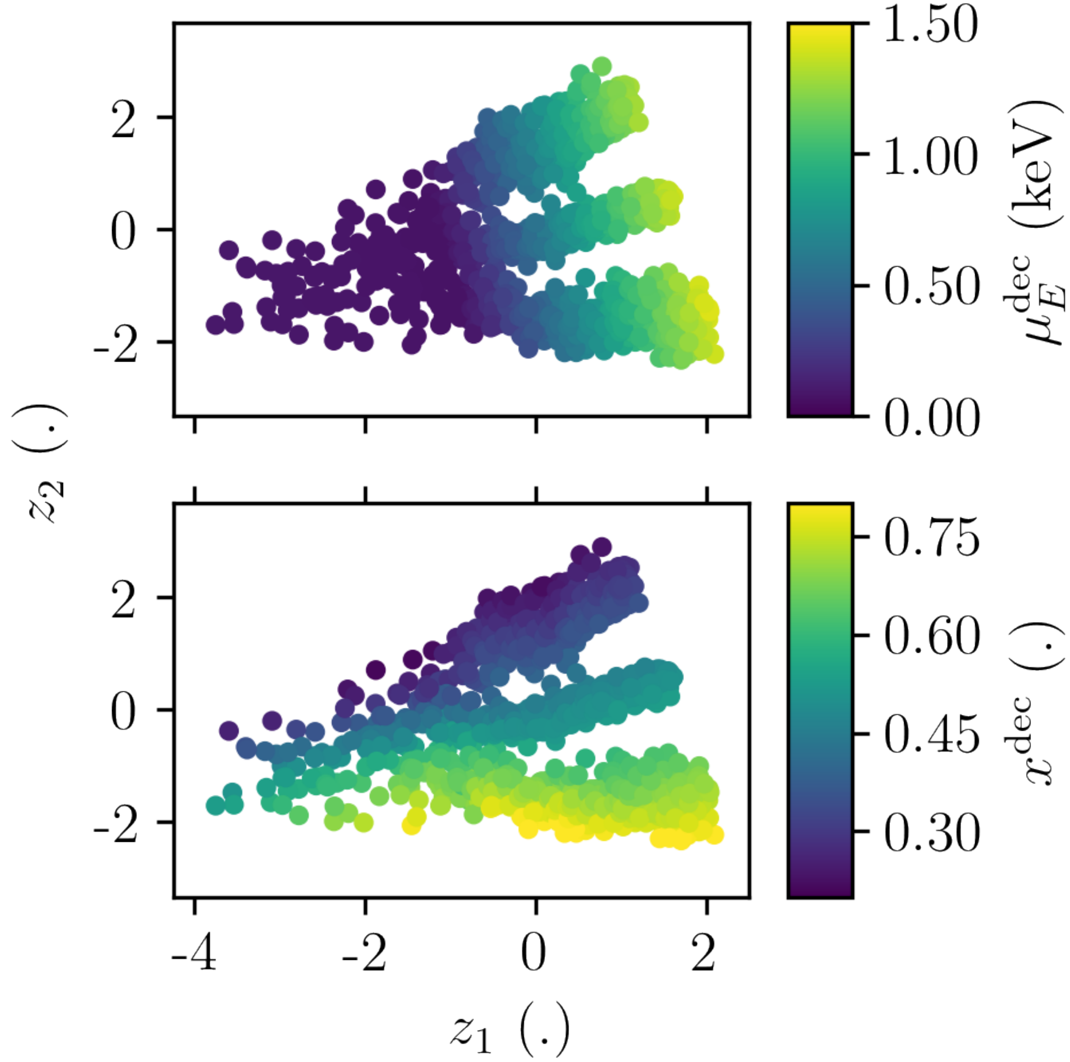


VAE with regression of mean ion energy $\langle E \rangle$ and stoichiometry x

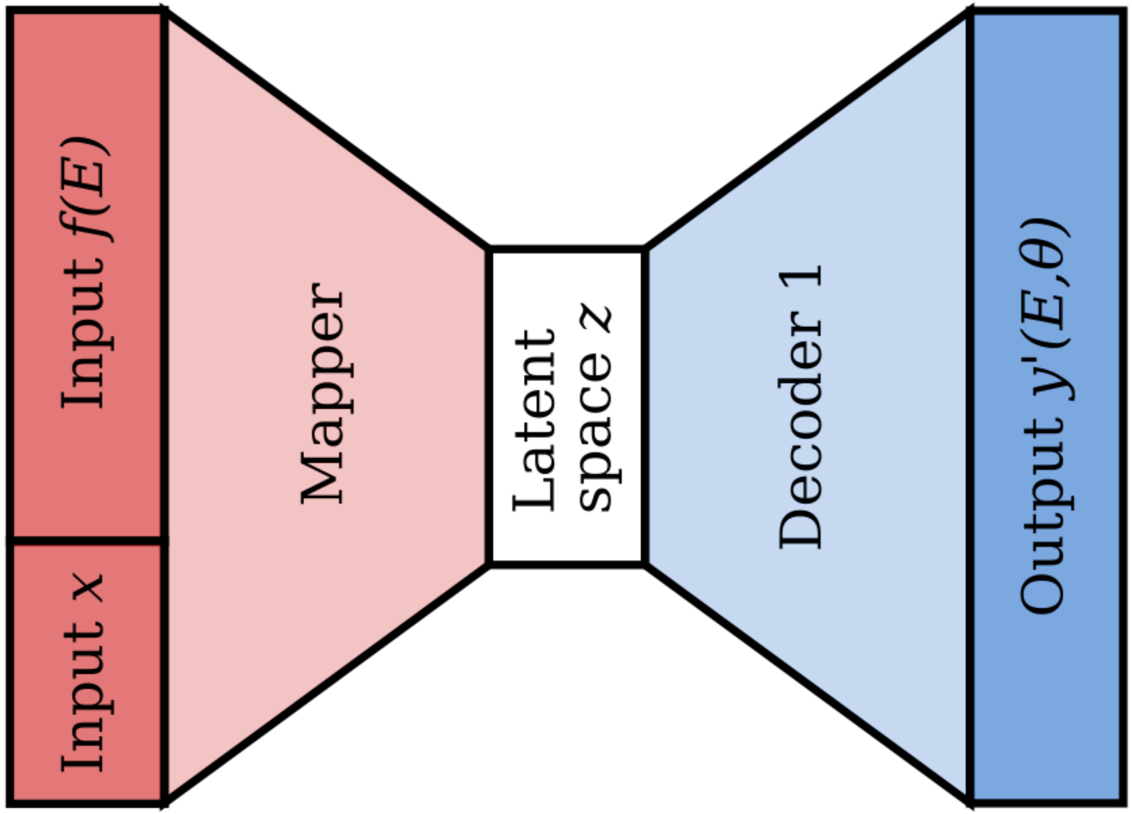
---

Artificial neural network simulated using Keras and TensorFlow, `www.tensorflow.org`
G.E. Hinton and R.R. Salakhutdinov, Science 313, 504 (2006)

2D latent space representation (possibly for EAD generation)



Mapper–decoder network and transfer learning

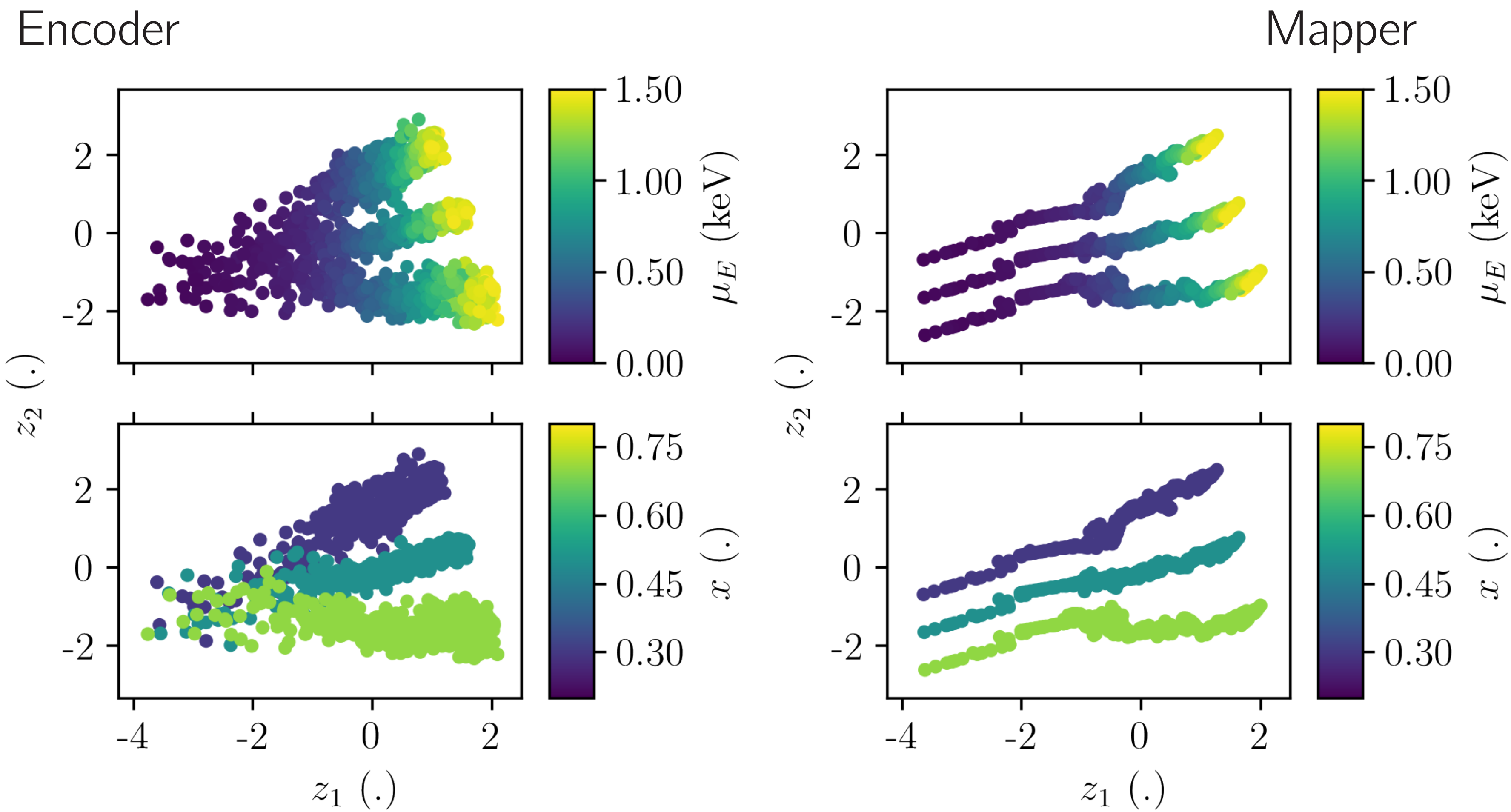


Mapper-decoder network utilizing previously trained VAE decoder

IEDF $f_{Ar^+}(E)$ and stoichiometry x as input

Artificial neural network simulated using Keras and TensorFlow, `www.tensorflow.org`
G.E. Hinton and R.R. Salakhutdinov, Science 313, 504 (2006)

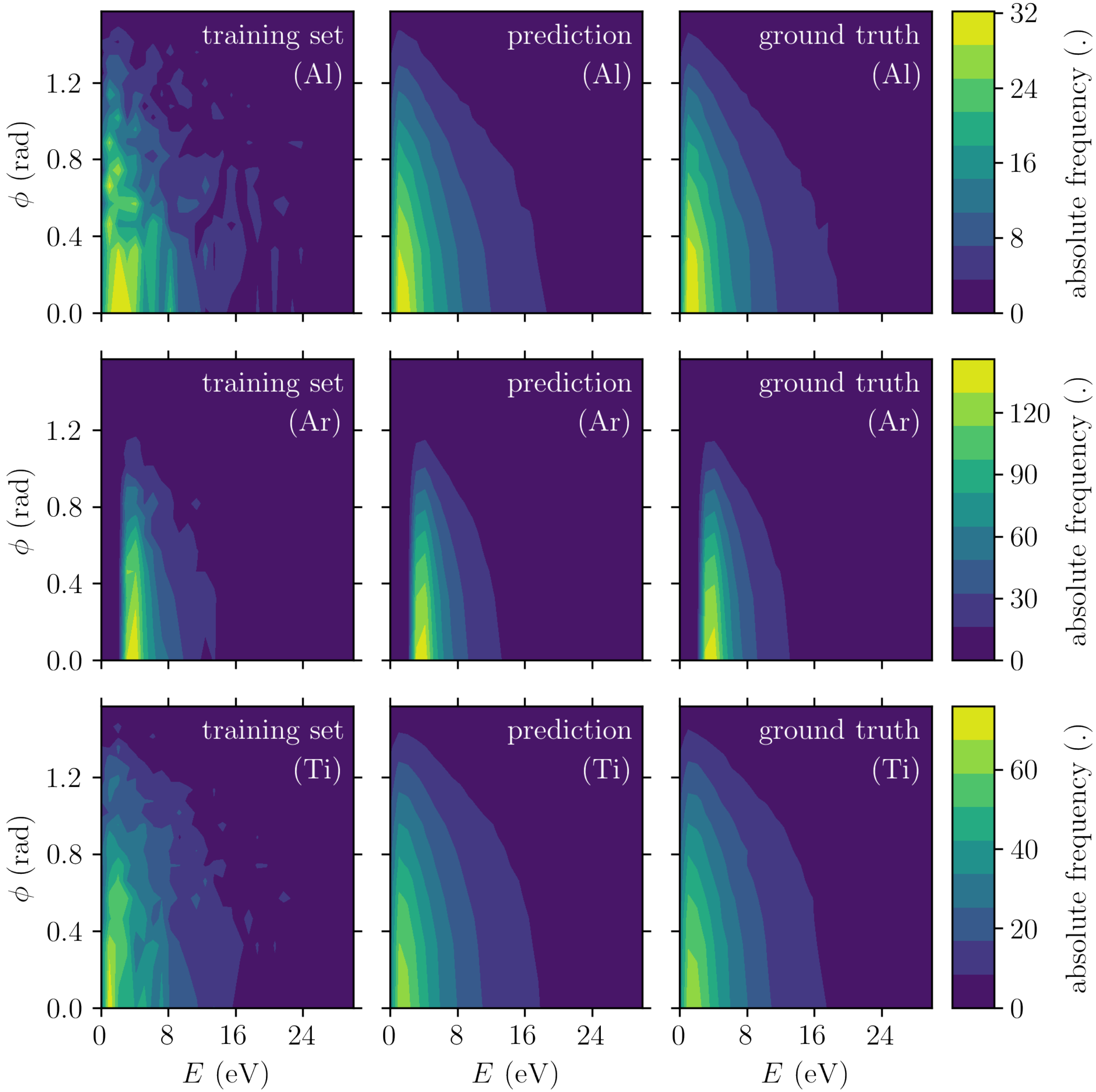
2D latent space representation



Mapper network distinguishes low ion energy / low sputter yield cases

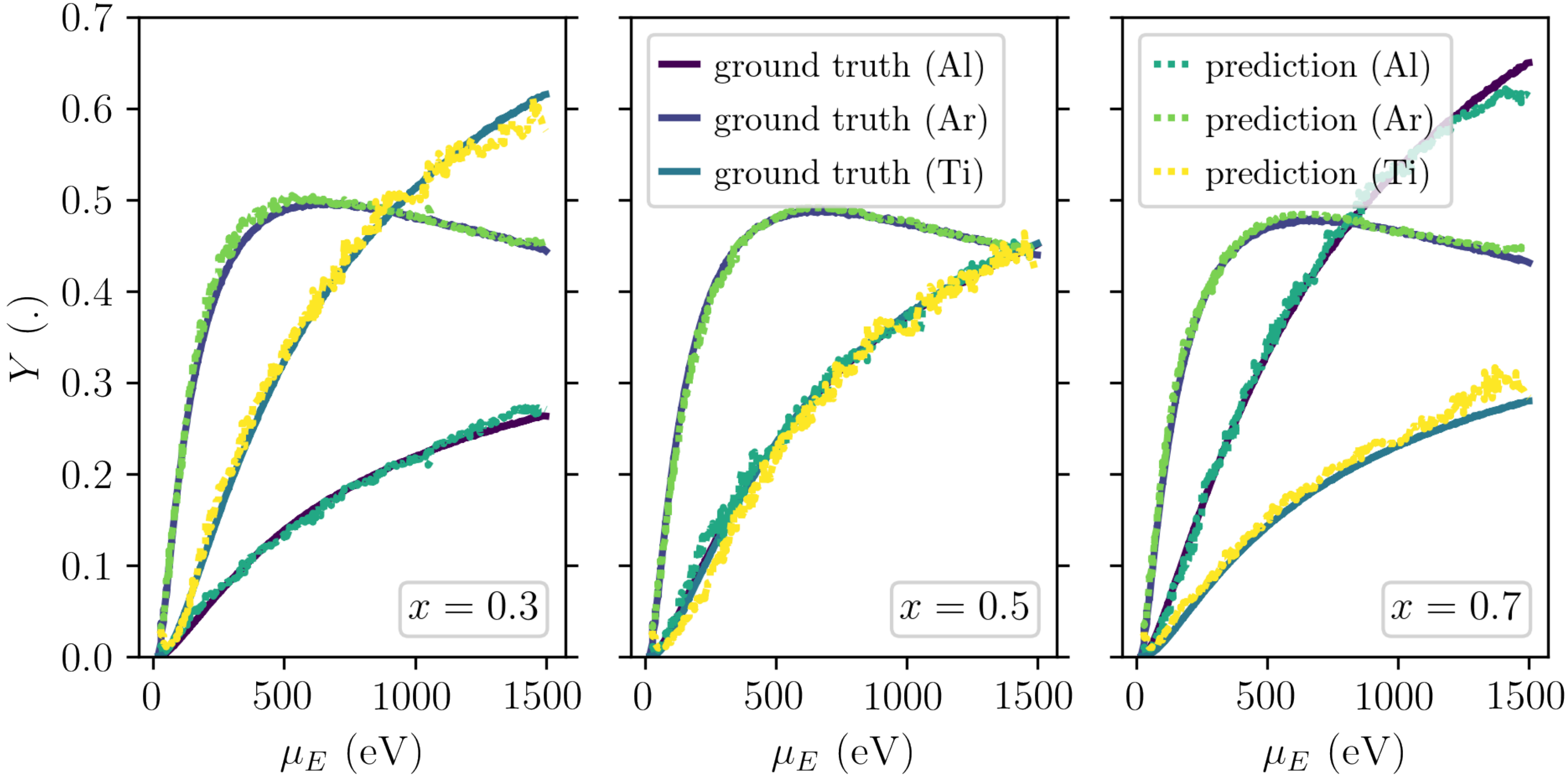
Latent space interpolation vs extrapolation

Prediction and generalization



$R^2 = 0.99$ over complete ground truth data set

Prediction: Integrated quantities



Physics preserved also for integrated quantities across data set

How complex need this network to be?

Fully connected network structure had a total of 3,955,800 weight parameters

Final optimized convolutional mapper–decoder network:

- 358 mapper weight parameters (regression training)
- 18,135 decoder weight parameters (autoencoder training)

Total of 18,493 weight parameters

Reduced **2-dimensional parameter space** provides sufficient latent representation

Concluding remarks

- Machine learning and artificial neural networks
  - Fundamental aspects
  - Machine learning surrogate models and physics-informed representation
- Chronological example: Plasma-surface model interface
  - Fully-connected artificial neural network with 3,955,800 parameters
  - Dimensionality reduction for TRIDYN data (Ar/Ti-Al **variable**)
  - Regression with 358 mapper and 18,135 decoder parameters
- Diverse applications in LTP modeling as well as experiments envisioned

SFB-TR 87

Gefördert durch DFG Deutsche Forschungsgemeinschaft