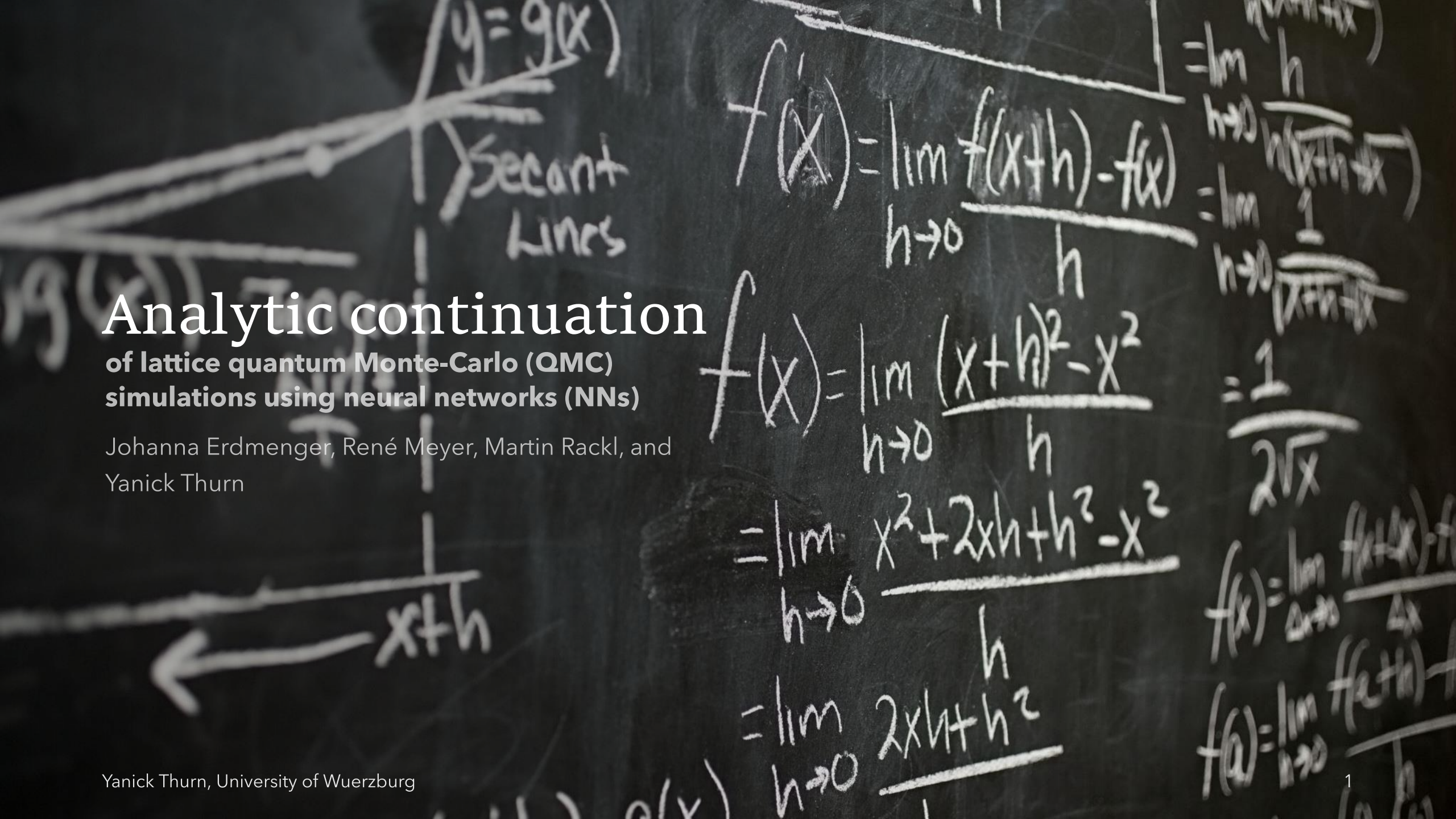


Analytic continuation

of lattice quantum Monte-Carlo (QMC) simulations using neural networks (NNs)

Johanna Erdmenger, René Meyer, Martin Rackl, and Yanick Thurn

Roadmap

- Motivation for the machine learning approach
- Introduction to neural networks
- Description of the physical problem we want to solve
- Computational setup and network architecture
- Results and comparison of the network to different methods
- Extraction of knowledge from the network

Motivation

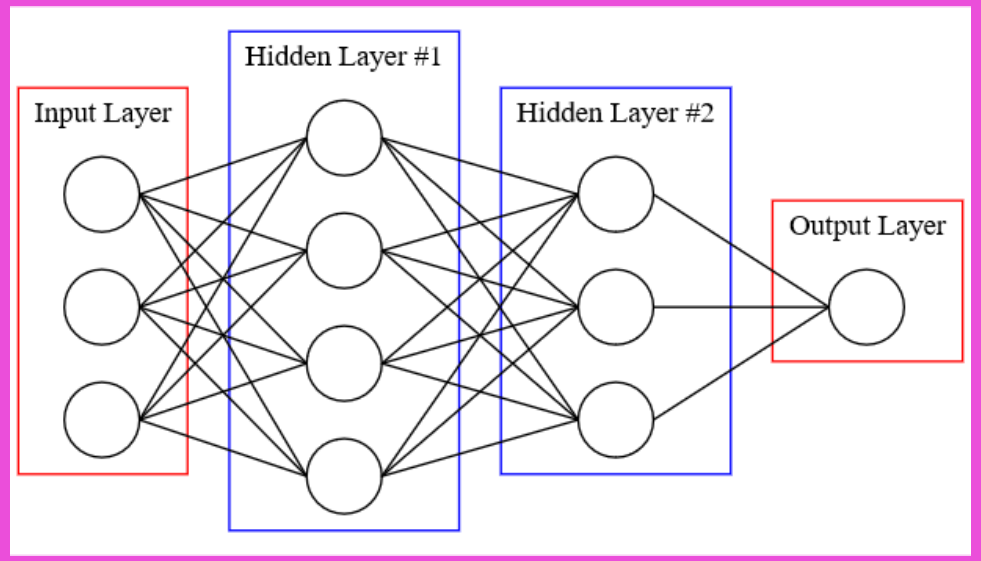
- Simulate behavior of fermions in solid states using Quantum Monte Carlo Simulations [arxiv:2012.11914]
- The results of these simulations are in the imaginary time domain and thus not comparable with reality. Their relation is ill-posed
- Idea: Using a neural network to construct the corresponding spectral density function [arXiv:1612.04895, arXiv:cond-mat/0612233, arXiv:1806.03841, arXiv:2302.11317, arXiv:2111.12266]
- Using a simple toy model (1D Hubbard Model) to test and develop a network for this task

Introduction to Neural Networks

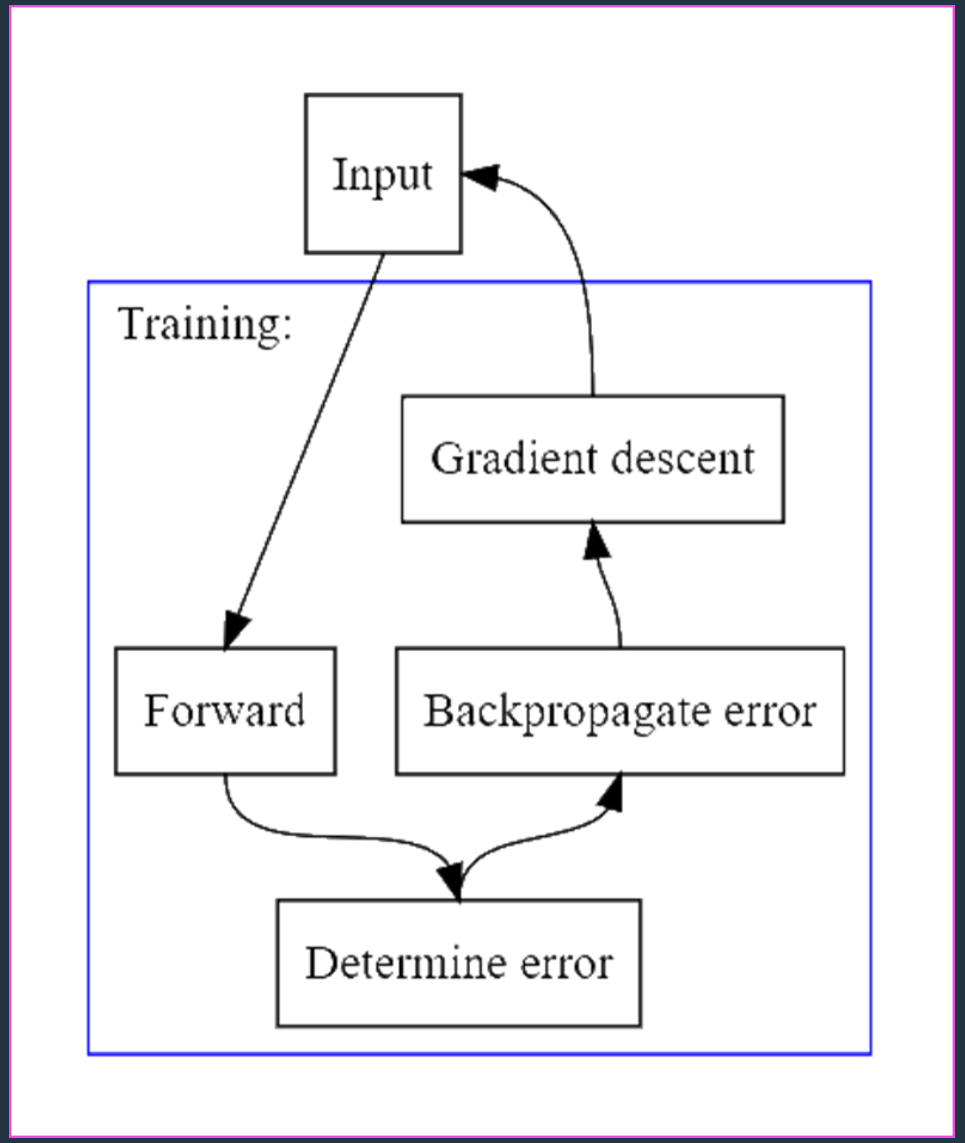
Neural networks in short

- Neural networks are networks of repeating units called neurons
- Universal function approximator $f: \mathbb{R}^N \rightarrow \mathbb{R}^M$
- Let $\vec{x} \in \mathbb{R}^N$ be an input, $W \in \mathbb{R}^N \times \mathbb{R}^M$ a weight matrix, f a non-linear activation function, and $\vec{b} \in \mathbb{R}^M$ the bias

$$\vec{y} = f\left(W \cdot \vec{x} + \vec{b}\right)$$



Backpropagation



The problem

The analytic continuation problem

$$\mathcal{G}(\tau) = -\int d\omega \frac{e^{-\omega\tau}}{1 + e^{-\frac{\hbar\omega}{k_B T}}} A(\omega) = \mathcal{K} \circ A(\omega)$$

With (Green und spectral function)

$$A(\omega) = \frac{1}{\pi} Im\left(G^R(\omega)\right)$$

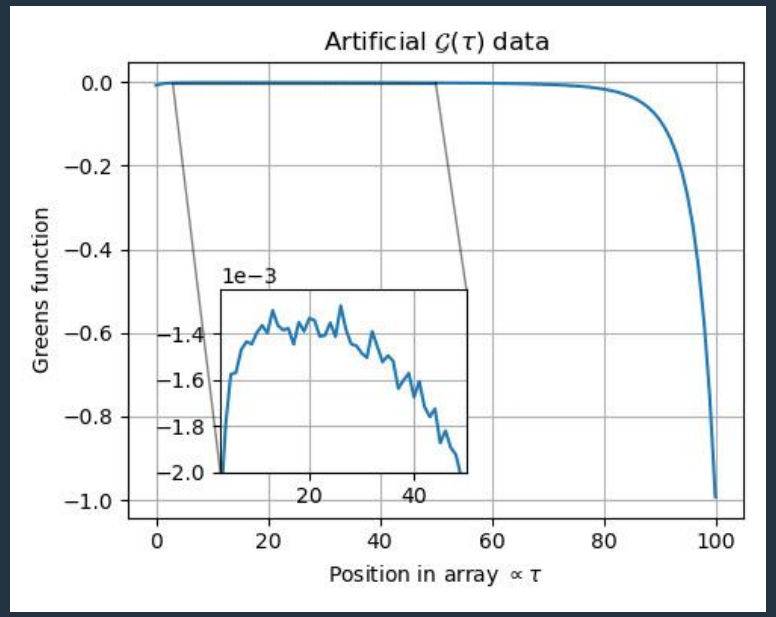
The analytic continuation problem

- As

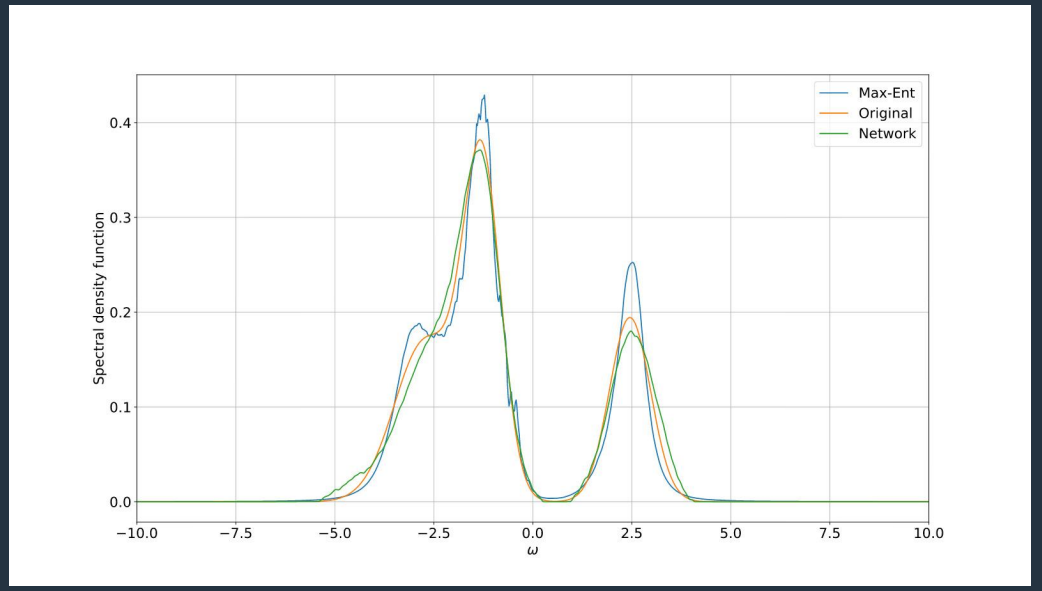
$$\lim_{\omega \to \pm\infty} \mathcal{K}(\omega) \to 0$$

Therefore, the inverse function reacts chaotically to noise

A direct solution for $A(\omega)$ is not stable



Green's function in imaginary time domain



Original (orange), Max-Ents (blue) and Networks (green) approximated spectral density function $A(\omega)$

(2020) Romain Fournier et al.

Computational aspects

Training data

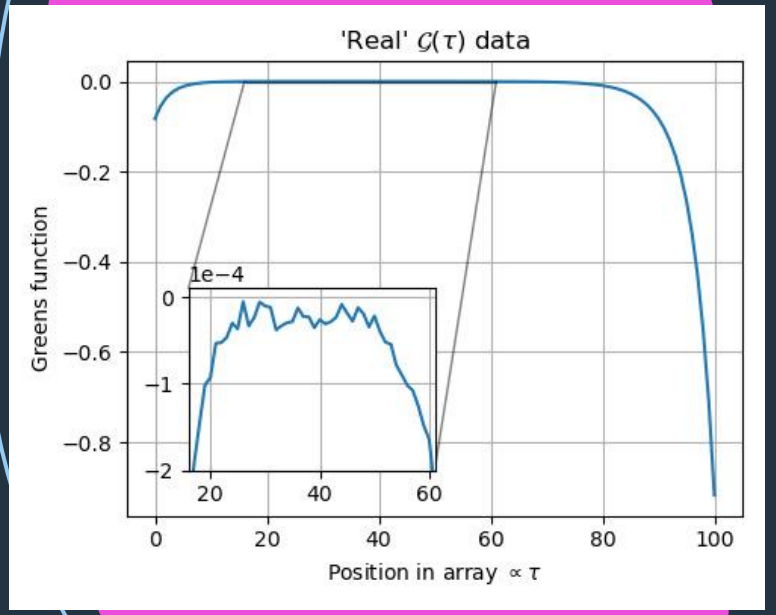
- We train a network in supervised manner using the Dataset

$$\mathcal{D} = \{(G(\tau), A(\omega)), \dots\}$$

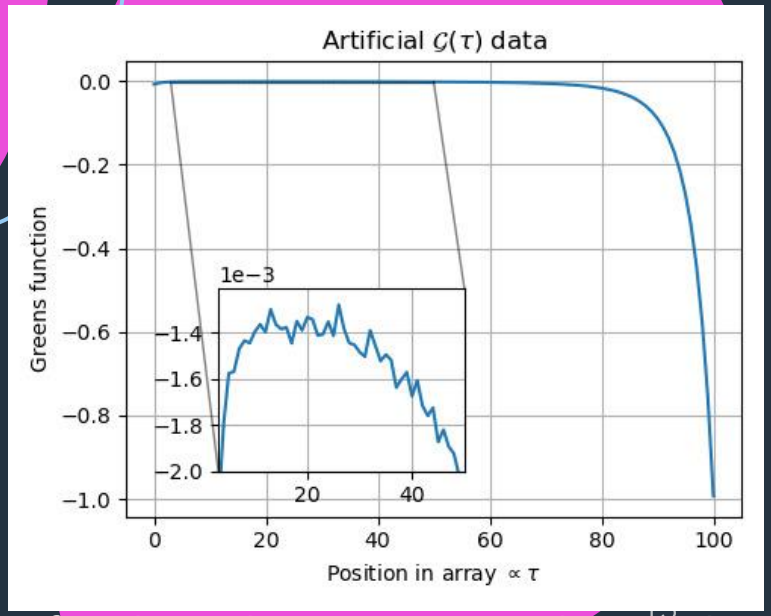
- The training data are randomly generated:
  - Select a number $n \in [1, 8]$ of gaussians (uniform random)
  - Place the gaussians randomly (uniform) at $\omega \in [-10,\ 10]$ with random variance $\sigma \in [0.4, 1.4]$
  - Select random height but with exponential decay towards boundaries
  - Normalize function (this is now the spectral density function)
  - Use $\mathcal{G}(\tau) = \mathcal{K} \circ A(\omega)$ to calculate the Greens function in the imaginary time domain

Overfitting

To prevent overfitting, the generated training get a new noise every epoch.



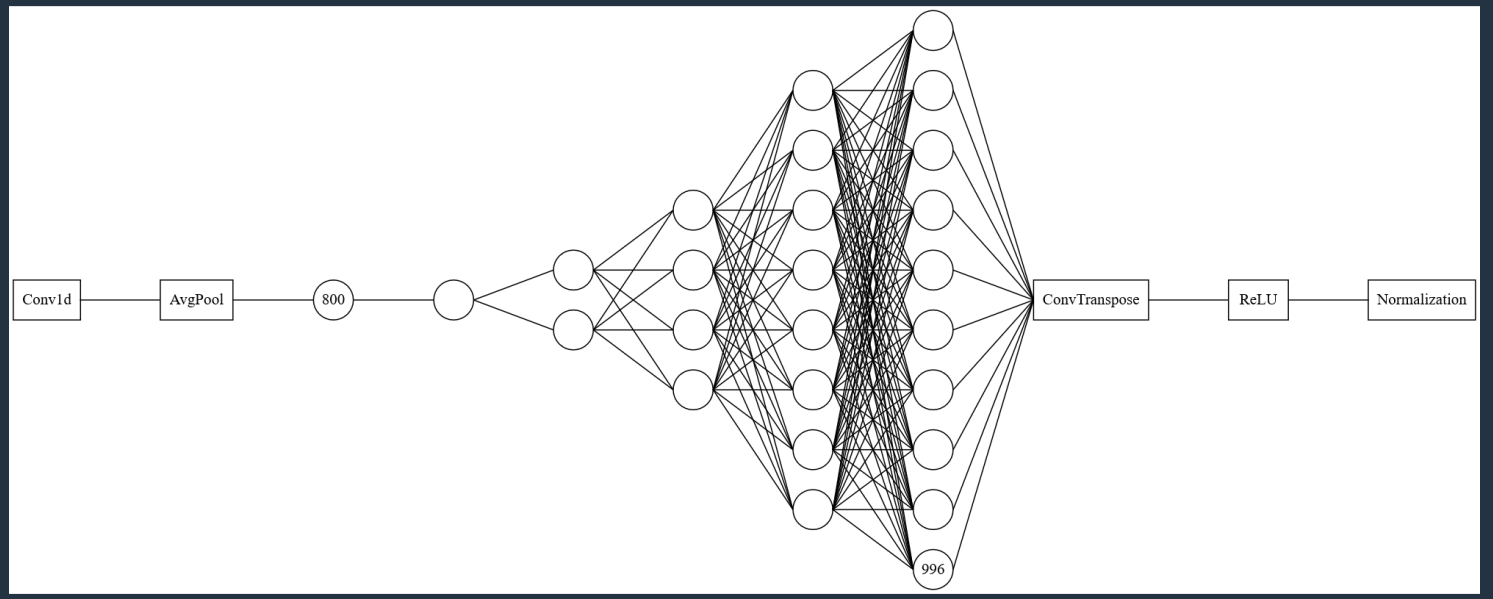
Real QMC $\mathcal{G}(\tau)$ data



Artificial random generated $\mathcal{G}(\tau)$ data

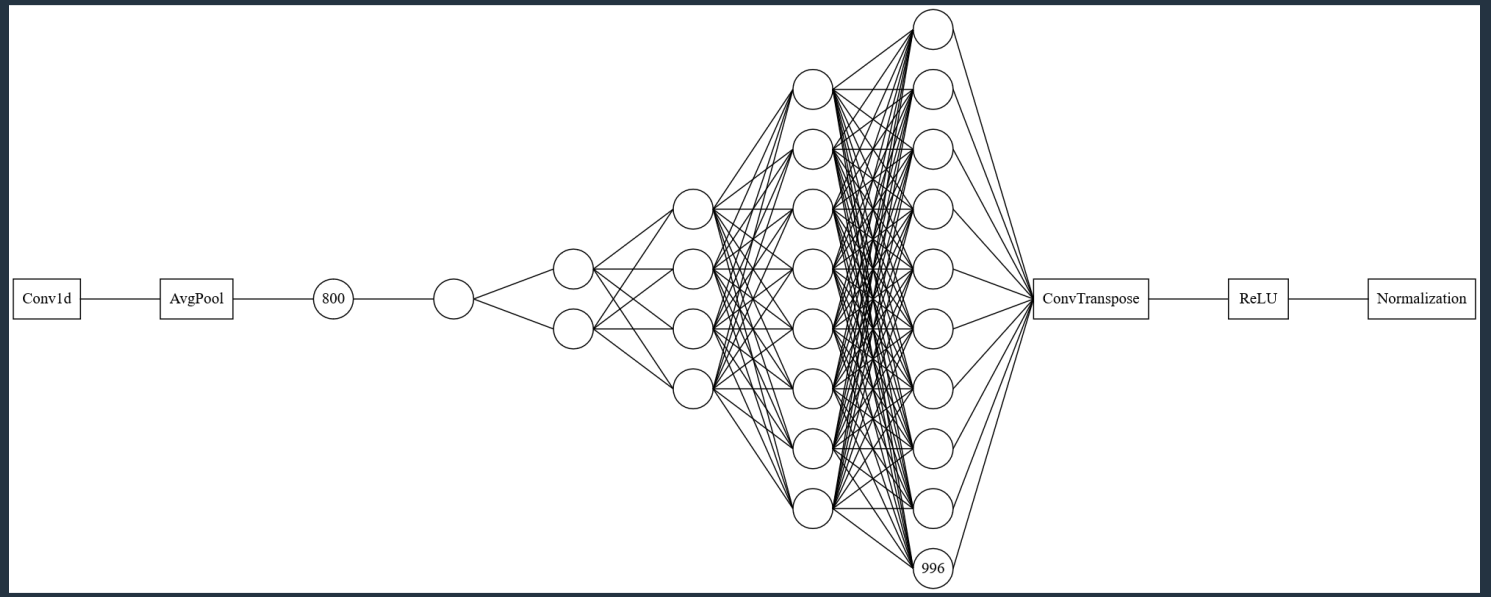
Yanick Thurn, University of Wuerzburg

Neural network



- The network consists of three parts: Convolutional and Dense
- The intuition for the convolutional part of the network is to allow it to compute local operations such as the gradient or fir filters

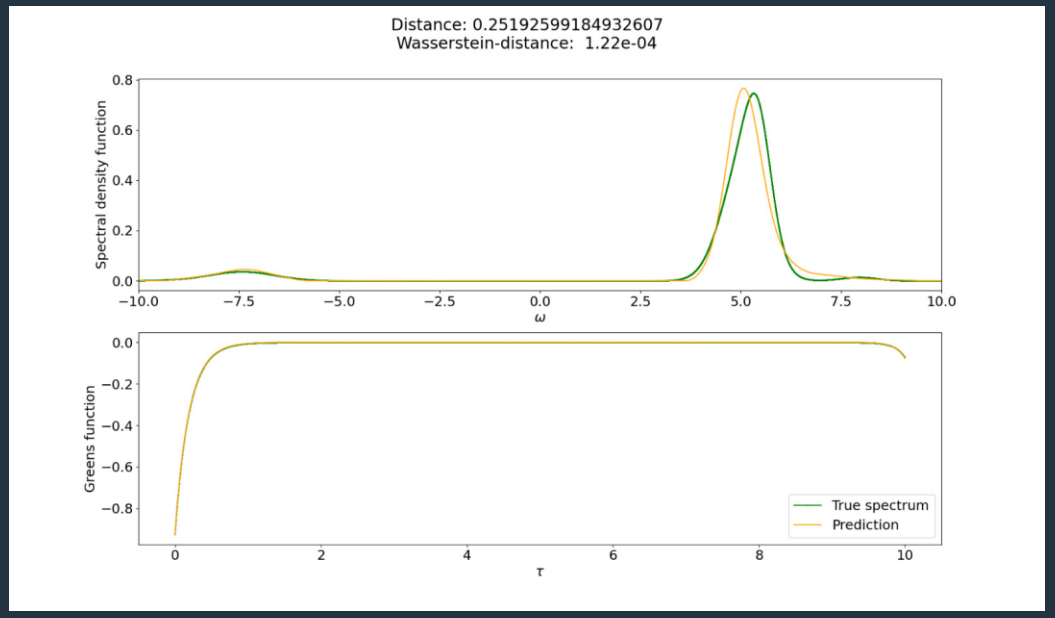
Intuition to our architecture

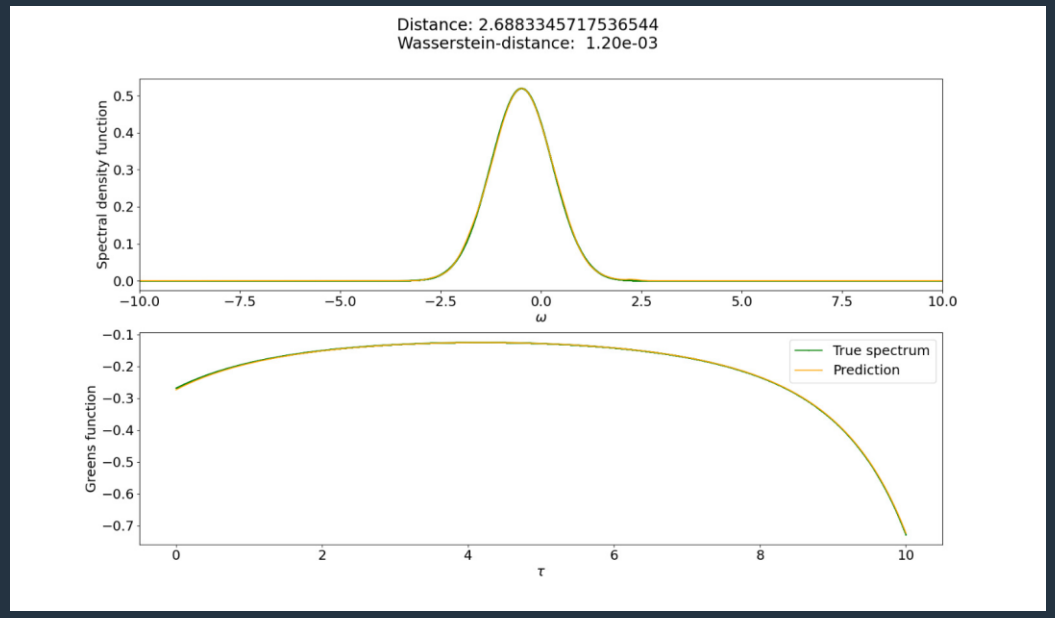


- Data include noise
  + local operators, i.e., slope, seem intuitively important ⇒ Conv. layer
- Symmetry of the data ⇒ AvgPool instead of MaxPool
- Output is not restricted ⇒ Divergent instead of Convergent activation function
- Negative output allowed ⇒ no ReLU (Moreover, its bad for training)
- Negative should not be treated different ⇒ no LeakyReLU
- Output requires many nodes to be precise ⇒ ConvTranspose
- Physical restrictions (Positive semi definite, Normalized) ⇒ ReLU + Normalization

Results

Example Images

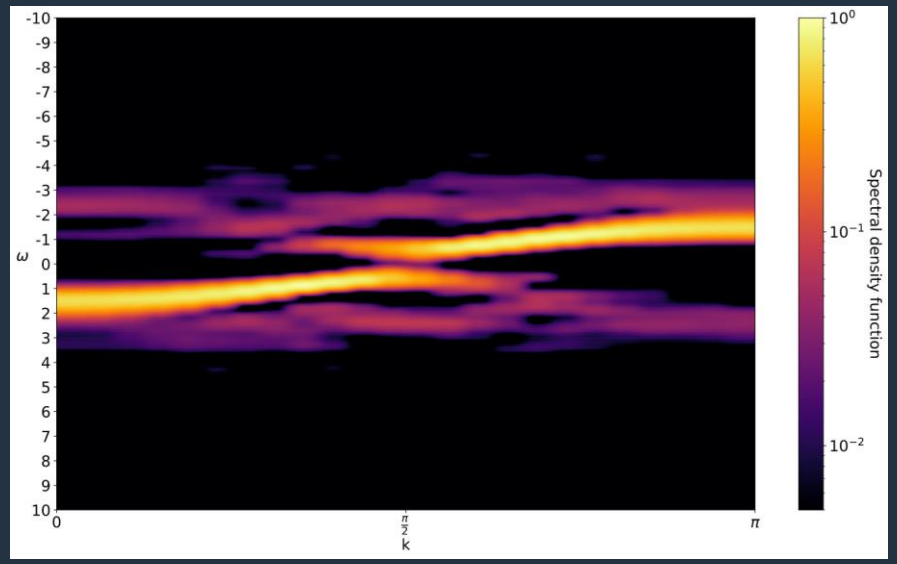




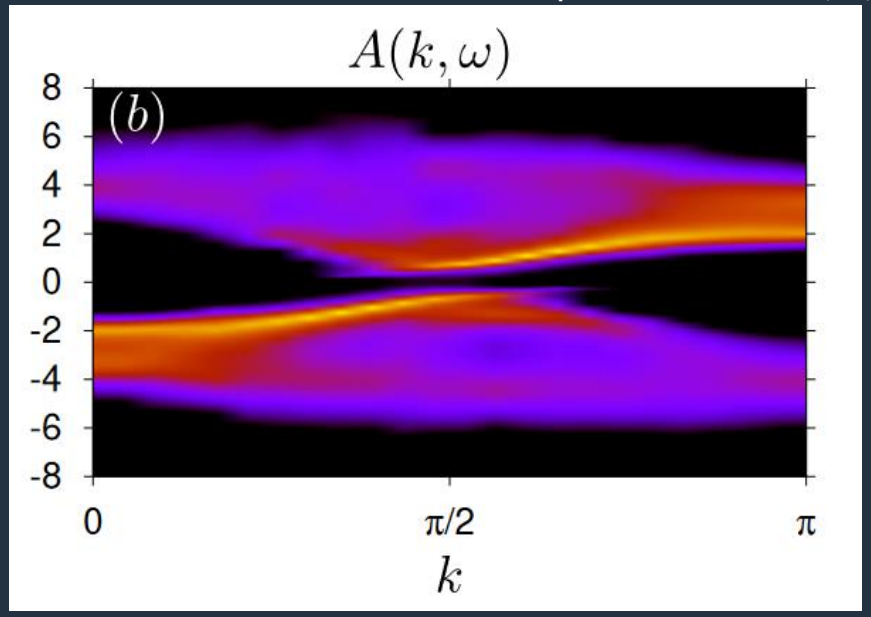
Spectral density function $A(\omega)$ [top] and Green's function in imaginary time domain $\mathcal{G}(\tau)$ [bottom] for truth [green] and approximated [orange] functions

k-space Image

- Spin-charge separation in 1d Hubbard model

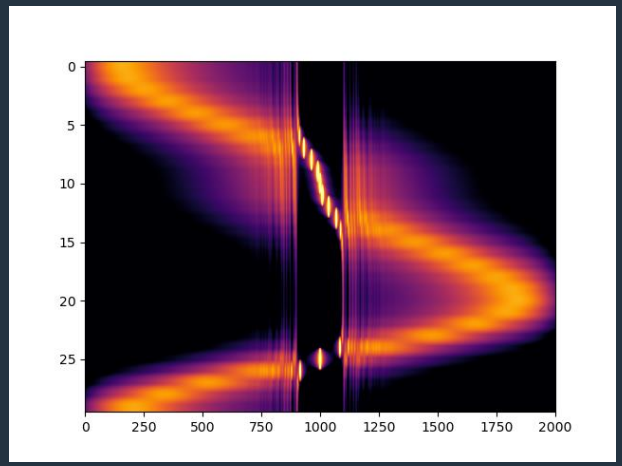


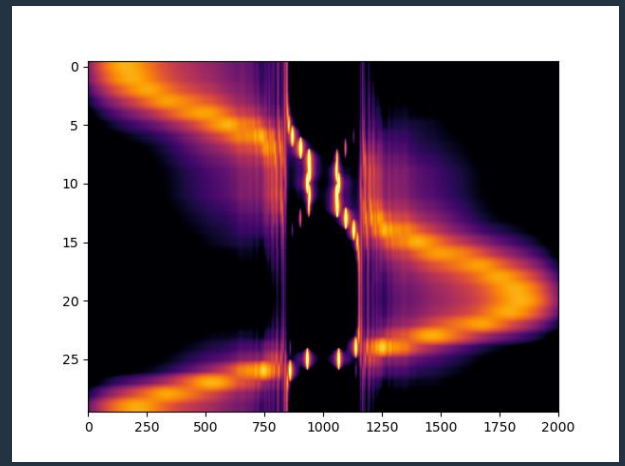
Predicted (network) $A(\omega)$ for different k.
The color encodes the amplitude of $A(\omega)$

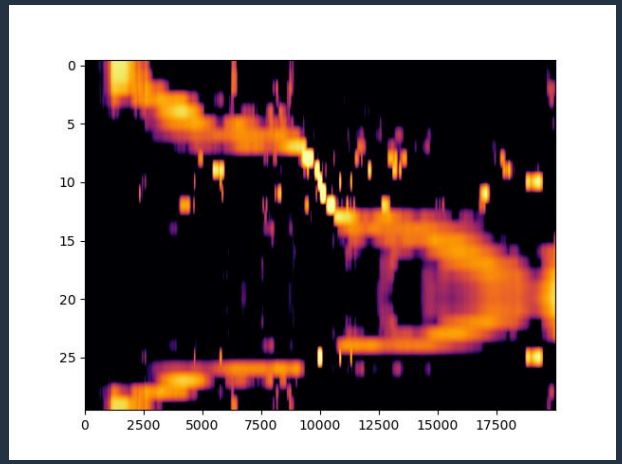


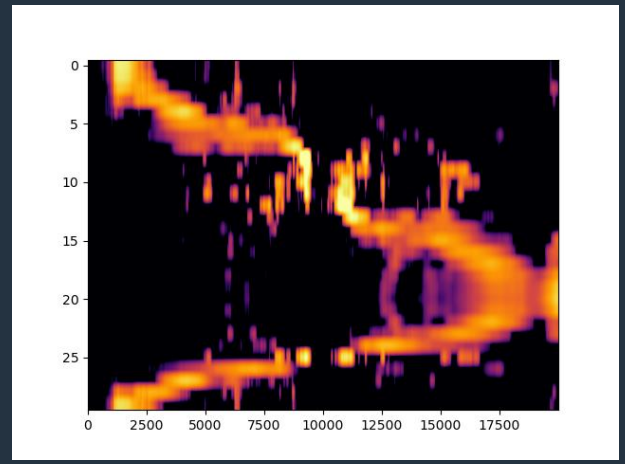
Predicted (Max-Ent) $A(\omega)$ for different k. The color encodes the amplitude of $A(\omega)$. (F.F. Assaad et al. 2012)

Comparison to analytic results



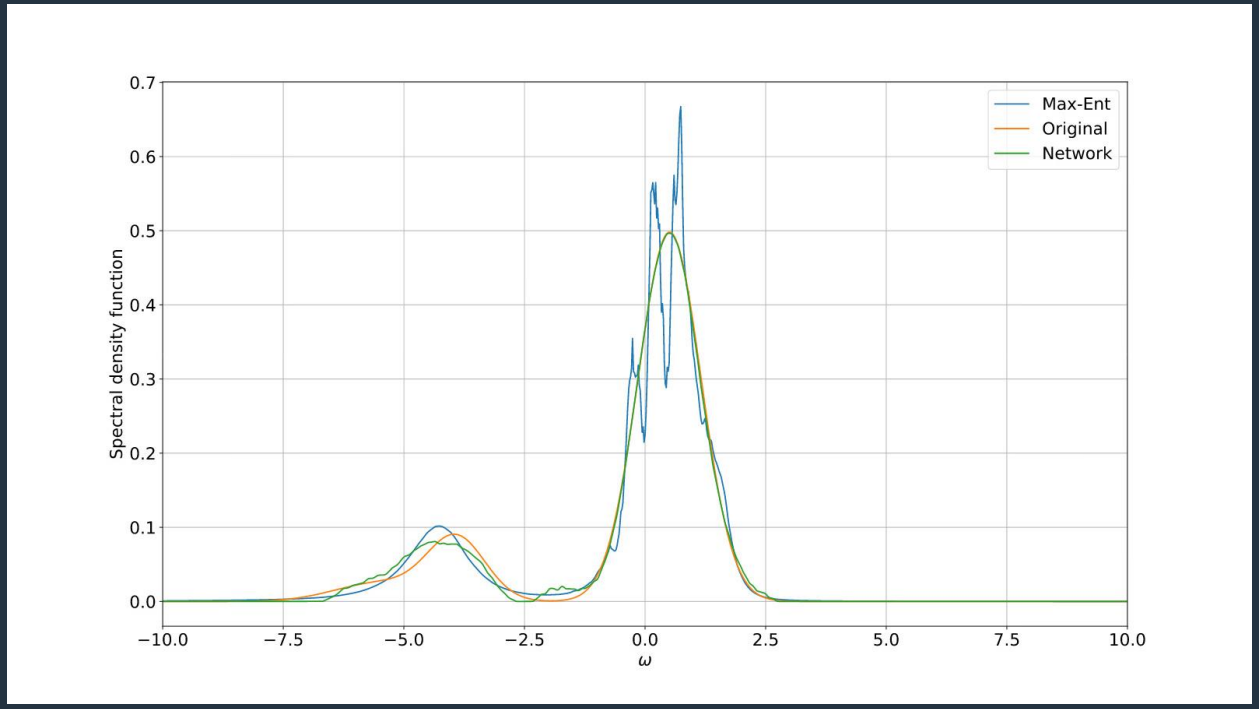




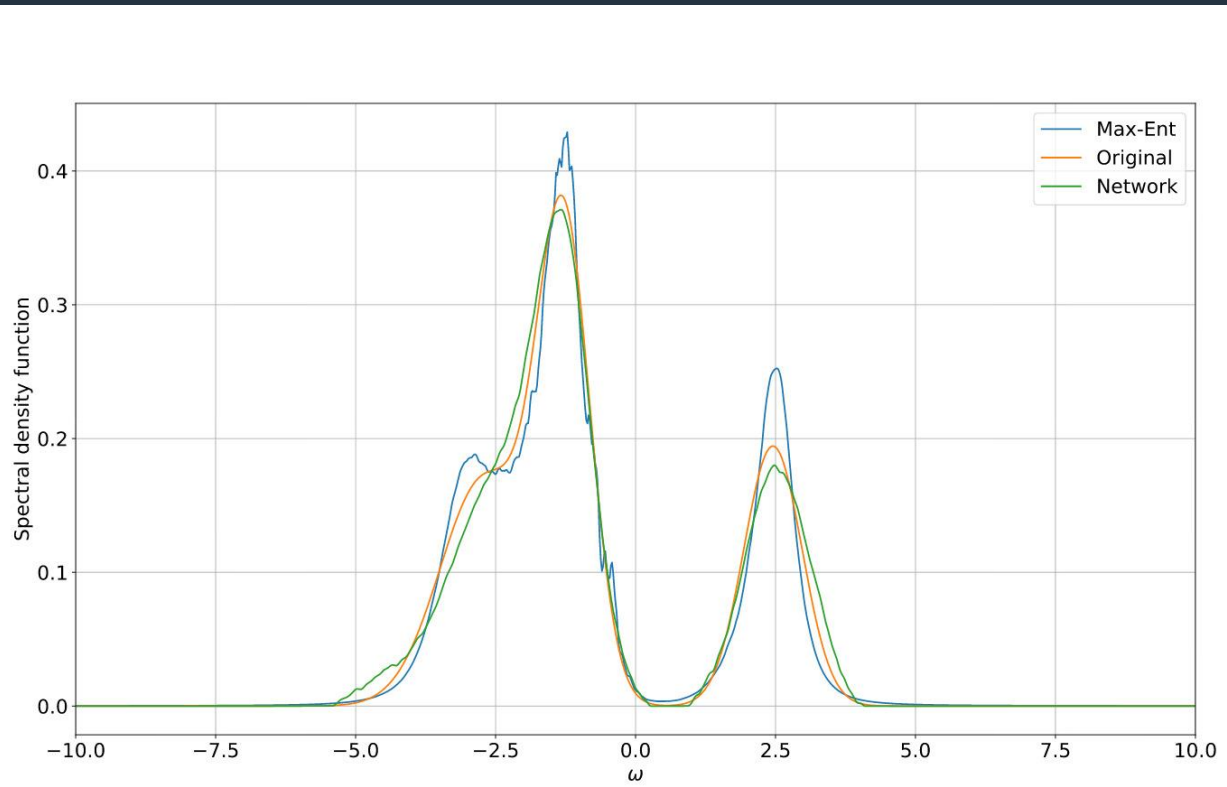


Comparison to Max-Ent

- Original (orange), Max-Ents (blue) and Networks (green) approximated spectral density function $A(\omega)$



Comparison to Max-Ent



- Original (orange), Max-Ents (blue) and Networks (green) approximated spectral density function $A(\omega)$

Comparison to Max-Ent

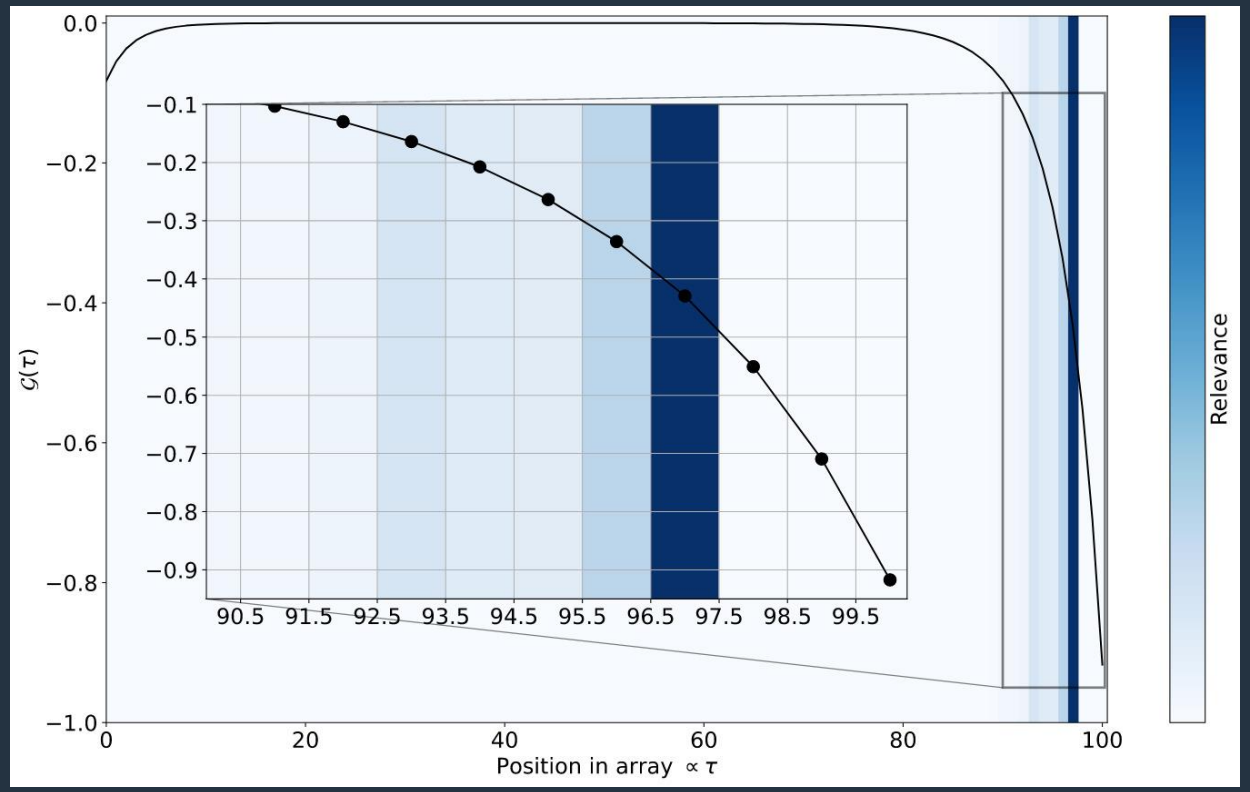
Over some (64) validation data, the difference between the predicted spectral density function and the expected one was

	Wasserstein distance	L1 Norm
Max-Ent	$(3.5 \pm 1.7) \cdot 10^{-3}$	(225 ± 94)
Network	$(2.0 \pm 1.2) \cdot 10^{-3}$	(125 ± 60)

What can we learn from this approach?

Layer-wise relevance propagation

- Usually used in categorization problems
- By summarizing over all output-nodes, we can see what the networks sees as relevant for its decision
- Not all parts of the Greens function are equally important
- The “longer” arm of the Greens function is more important



Relevance (blue colorbar) for different values of $\mathcal{G}(\tau)$ for the neural network

Conclusion & Outlook

- Using the network we were able to see the spin-charge separation in the k-space image
- For the tested Green functions, the predicted spectral density function had a distance to the expectation of
  - Max-Ent: $(3.5 \pm 1.7) \cdot 10^{-3}$ [Wasserstein], (225 ± 94) [L1-Norm]
  - Network: $(2.0 \pm 1.2) \cdot 10^{-3}$ [Wasserstein], (125 ± 60) [L1-Norm]
- In contrast to Max-Ent, the network does not rely on a priori knowledge of the spectral density function.
- At least for the network, it seems the input points of the Greens function are not equally relevant. Instead, certain parts of the function seem to be of special relevance
- If the network can be improved further, it may be a suitable alternative for Max-Ent, if no a priori knowledge is available
- Understanding how the network weights the points may yield improvements for Max-Ent

Hubbard model

- Simple model for electrons in solids
- Simulation on a lattice
- Electrons on the same slot experience Coulomb interactions
- Electrons behavior is described by three aspects:
  - Coulomb repulsion
  - Kinetic energy
  - Pauli principle