Interpretable tensor-neural networks as topological invariants

"Why is topology hard to learn?"

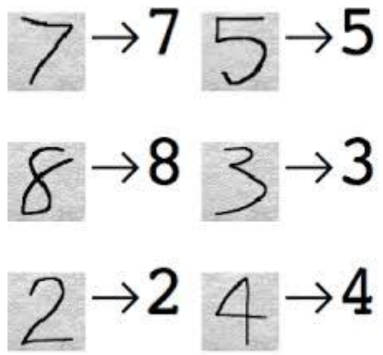
arxiv: 2509.26261

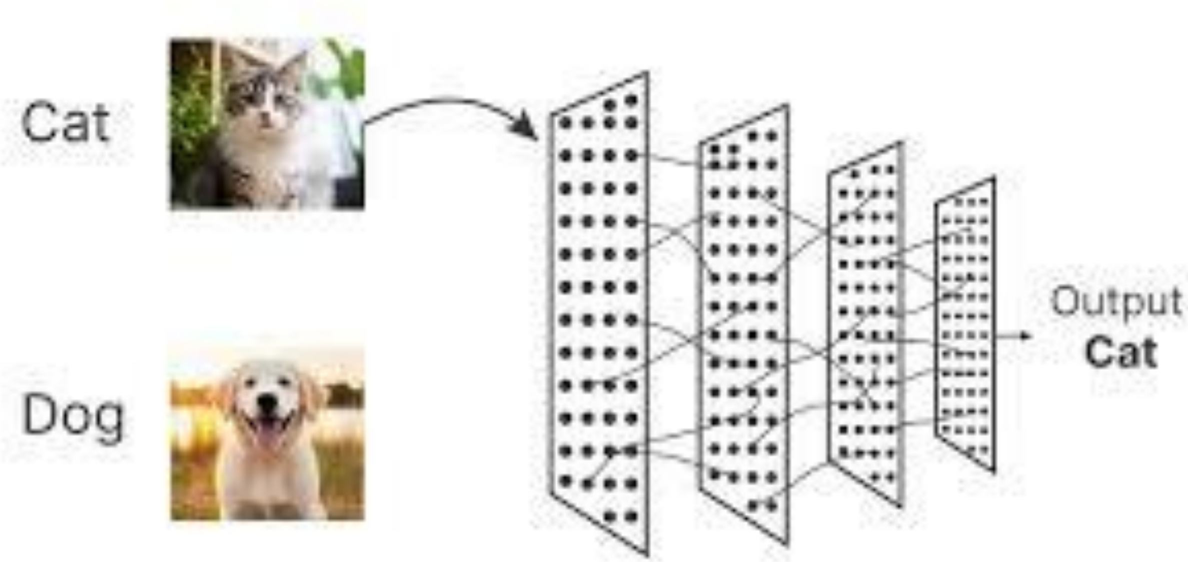
(and a bit of arxiv:2411.17822)

Dmytro Oriekhov

QMAI group, Kavli Institute of Nanoscience and QuTech, TU Delft

Where the neural networks started success path: Classifying cats, dogs, handwritten digits, ...



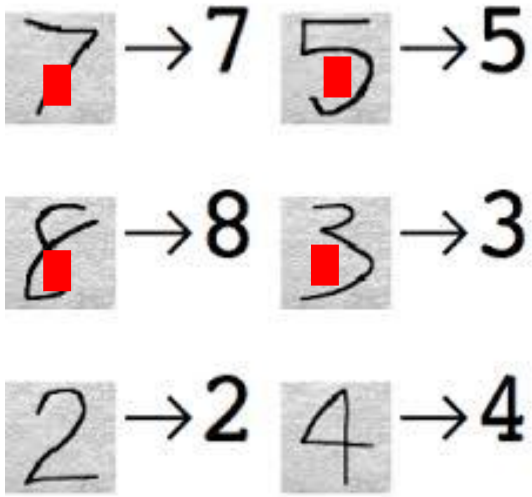


Nowadays: chatGPTs, Large Language Models

Classifying topological phase (index) from images… why is it hard?

- Requires global look at the system
- Should avoid "bit by bit comparison"

- Bit by bit works for digits very well!



The task of this study:

how to combine a powerful technique that “likes to zoom” with a global characteristic of a system?

$$\int_{BZ} dk \; \ldots..$$

$$\sum_{spectrum, X=0,..L} \ldots \ldots.$$

- Robust under “deformations”
- Not sensitive to disorder
- Universal

$$\nu = \frac{1}{2\pi i}\int_{BZ} \mathrm{Tr}\{[h(k)^{-1}\partial_k h(k)]\} \in \mathbb{Z}.$$

. P. Schnyder, S. Ryu, and A. W. Ludwig, PRL , 196804 (2009)

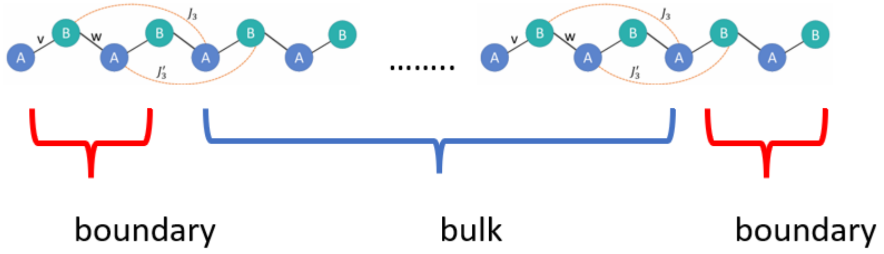
A bit of technical intro’s

Target systems – topological insulators

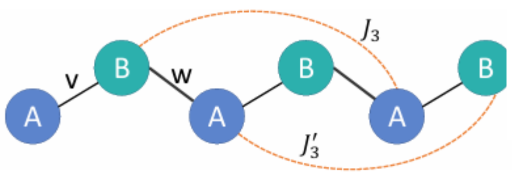
Example on which we focus: AIII (most typically BDI class),

Target regime - 1D real space (what we have in experiment)
Tunable hopping as "deformation"

Specific examples – SSH type models

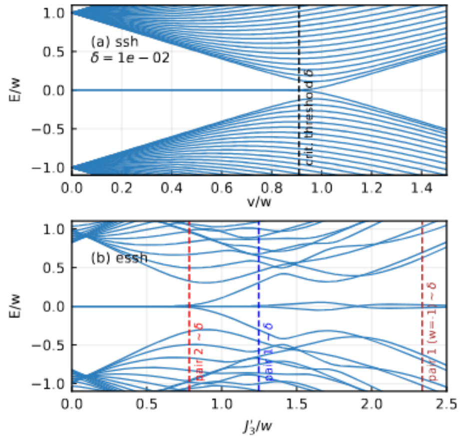


Physics of SSH family models



$$H=\sum_{n=1}^{N}\Big(vc_{n,A}^{\dagger}c_{n,B}+wc_{n,B}^{\dagger}c_{n+1,A}$$
$$+J_3c_{n,B}^{\dagger}c_{n+2,A}+J_3'c_{n,A}^{\dagger}c_{n+1,B}\Big)+\text{ h.c.},$$

$$H(k)=\begin{pmatrix}0 & h(k)\\ h(k)^{\dagger} & 0\end{pmatrix}$$

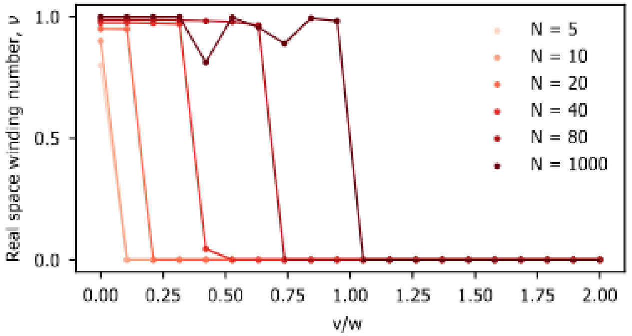


Chiral symmetry $\{\Gamma, H\}=0$

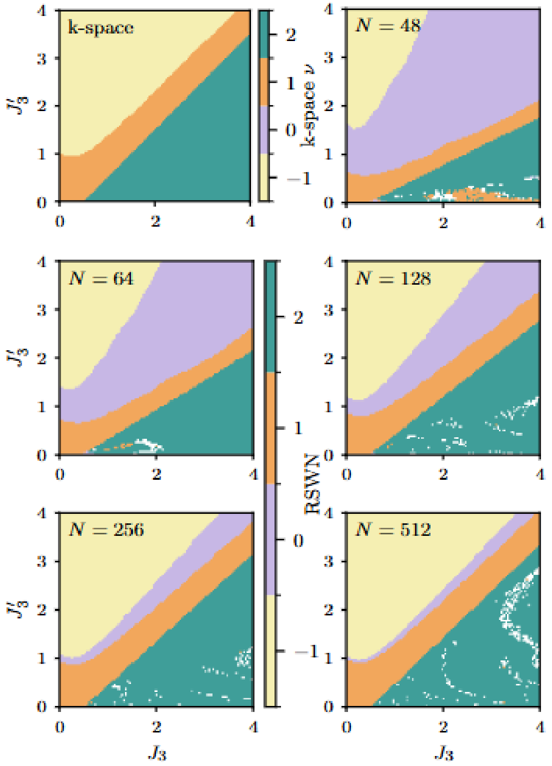
T-symmetry

Inversion symmetry

$$\nu=\frac{1}{2\pi i}\int_{BZ}\mathrm{Tr}\{[h(k)^{-1}\partial_k h(k)]\}\in\mathbb{Z}.$$



J3,J3’=0 v=0.5w ->



Finite size phase diagrams
G. Jin, **DO**, LJS, EG,
arxiv:2411.17822

Real-space winding number

$$\nu=-\frac{1}{N}\mathrm{Tr}\left\{P_B Q P_A\left[X, P_A Q P_B\right]\right\}$$

$P_{A,B}$-sublatt. projector

$Q=P_{+}-P_{-}$ - projection on filled empty bands

X – coordinate

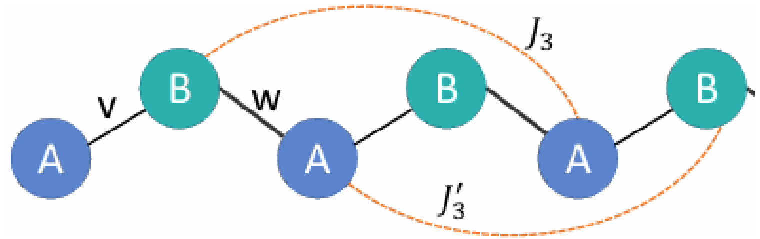
[X,] instead of k-derivative

W.-P. Su, J. R. Schrieffer, and A. J. Heeger,, PRL 42, 1698 (1979)

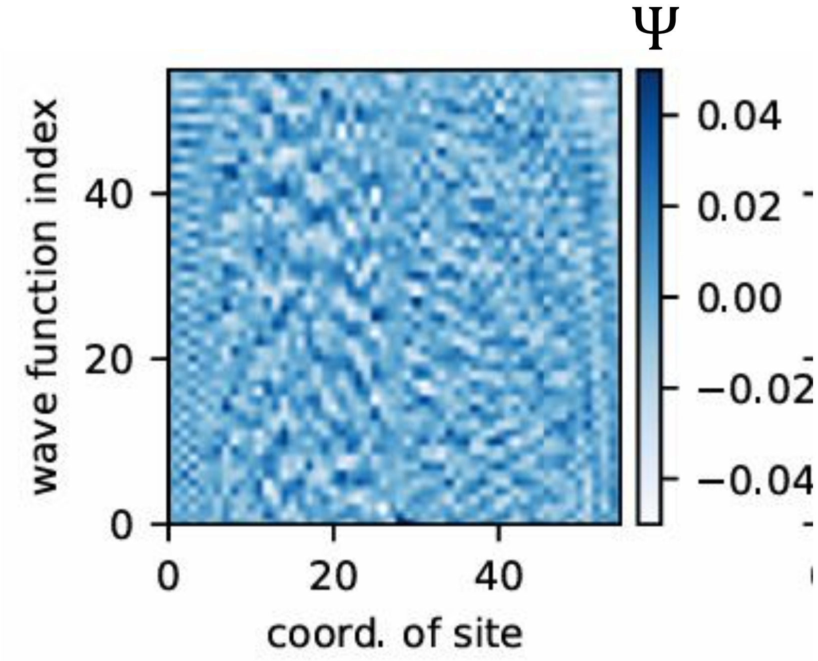
Ian Mondragon-Shem, Taylor L. Hughes, Juntao Song, Emil Prodan, PRL 113, 046802 (2014)

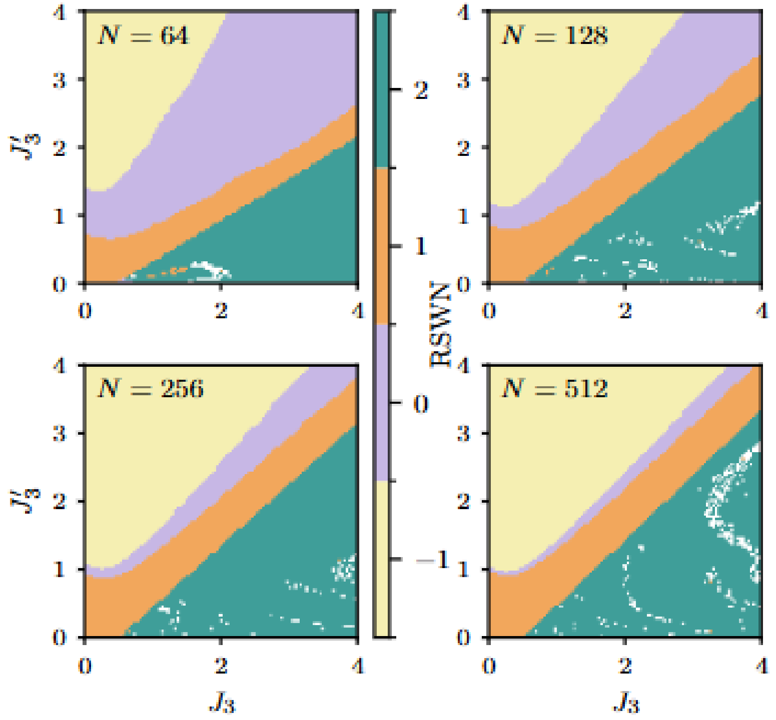
What do we train neural networks on?

- Images of all single-particle wave functions of 1D topological insulator, in real space



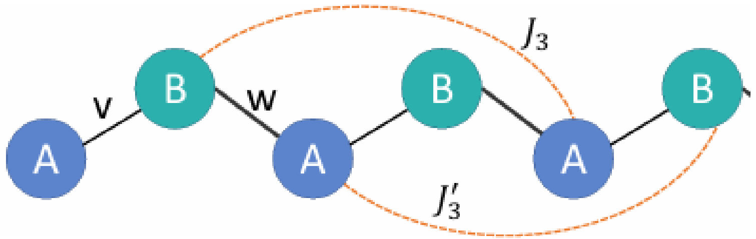
- Label – winding number



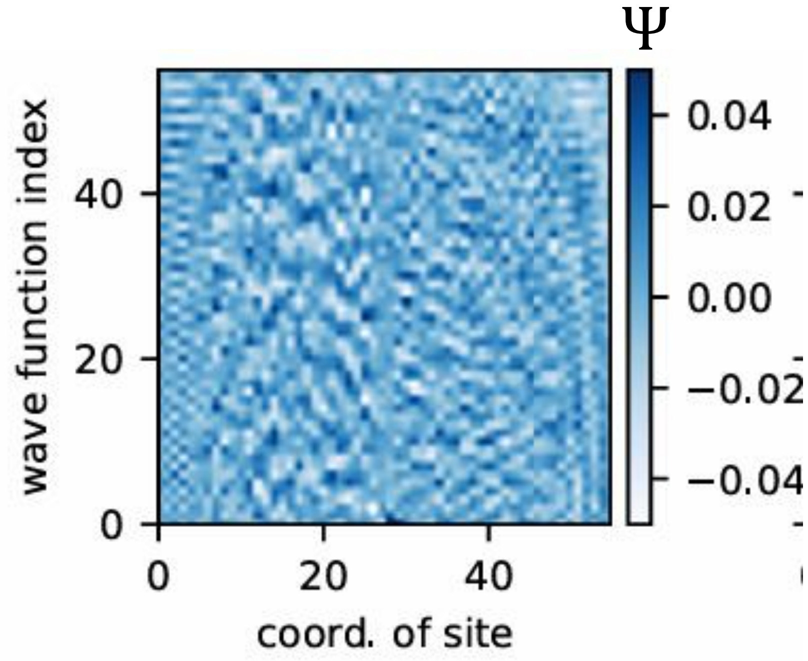


What do we train neural networks on?

- Images of all single-particle wave functions of 1D topological insulator, in real space



- Label – winding number



- SSH models have real wave functions
- Could be recovered from energy-resolved LDOS in experiment

Experiments at TU Delft: electrostatic control over the inductive intracell coupling via semiconductor nanowire junctions, for superconducting resonators
Gate-tunable phase transition in a resonator-based Su-Schrieffer-Heeger chain
L. Splitthoff et.al., PRR **6**, 043286 (2024)

Machine learning “black box” construction

Idea: topological invariant as neural network

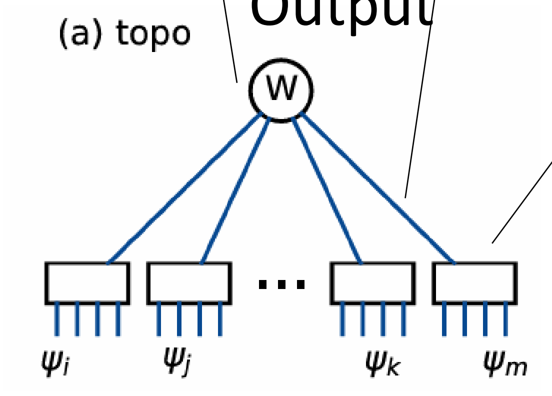
Real-space winding number is a sum of powers of inputs (wave functions): …

$$W = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{m=1}^{N} c(j,k)\left(X_{2m-1} - X_{2i}\right)\psi^{j}_{2i}\psi^{j}_{2m-1}\psi^{k}_{2i}\psi^{k}_{2m-1}, \quad \Longleftarrow \quad \nu = -\frac{1}{N}\mathrm{Tr}\left\{P_B Q P_A \left[X, P_A Q P_B\right]\right\}$$

$$c(j,k) = \mathrm{sign}\left(k - N - \frac{1}{2}\right)\cdot \mathrm{sign}\left(j - N - \frac{1}{2}\right).$$

…..As neural network!

Output



inputs

Just ML definitions:

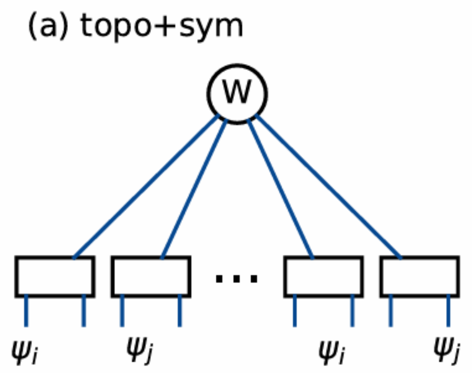
- Box (tensor network element) – multiplication of 4 inputs
- Circle connected to lines – sum of all inputs
- Lines – weights of neural network
- arxiv: 2509.26261

First run outcomes: trained neural network suggested existence of simplified formula

- Feature of SSH family of models; 2 sublattices
- E>0 and E<0 are linked via symmetry A,B->A,-B
- 2-point correlators are enough

$$\delta_{n,n'} = \langle \Psi^{n,+} | \Psi^{n',+} \rangle = \sum_x (\Psi^{n,+}_{x,A}\Psi^{n',+}_{x,A} + \Psi^{n,+}_{x,B}\Psi^{n',+}_{x,B}),$$

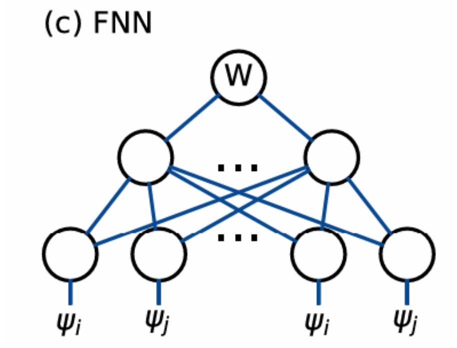
$$0 = \langle \Psi^{n,+} | \Psi^{n',-} \rangle = \sum_x (\Psi^{n,+}_{x,A}\Psi^{n',+}_{x,A} - \Psi^{n,+}_{x,B}\Psi^{n',+}_{x,B})$$



$$W = 2 \sum_{n \notin \{\text{edge st.}\}} \sum_x x \left(\left[\Psi^{n,+}_{x,A}\right]^2 - \left[\Psi^{n,+}_{x,B}\right]^2 \right) + \begin{cases} 2 \sum_{n \in \{\text{hybr. edge states}\}} \sum_x x \left(\left[\Psi^{n,+}_{x,A}\right]^2 - \left[\Psi^{n,+}_{x,B}\right]^2 \right), \\ 4 \sum\limits_{n \in \{\text{loc. edge st., B}\}} \sum_x x \left[\Psi^{n,+}_{x,A}\right]^2 - 4 \sum\limits_{n \in \{\text{loc. edge st., A}\}} \sum_x x \left[\Psi^{n,+}_{x,B}\right]^2 \end{cases}$$

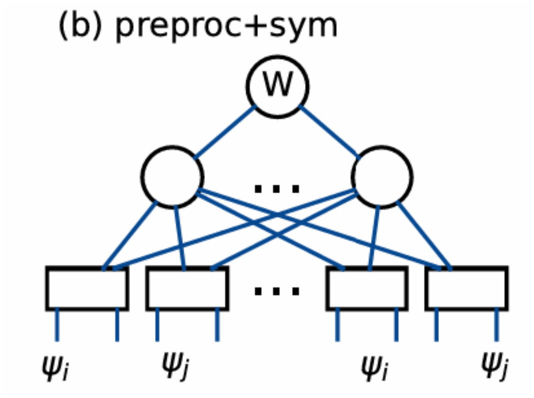
Training performance on SHH + ESSH dataset

Usual network

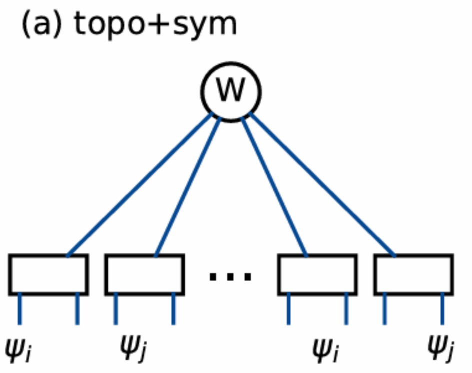


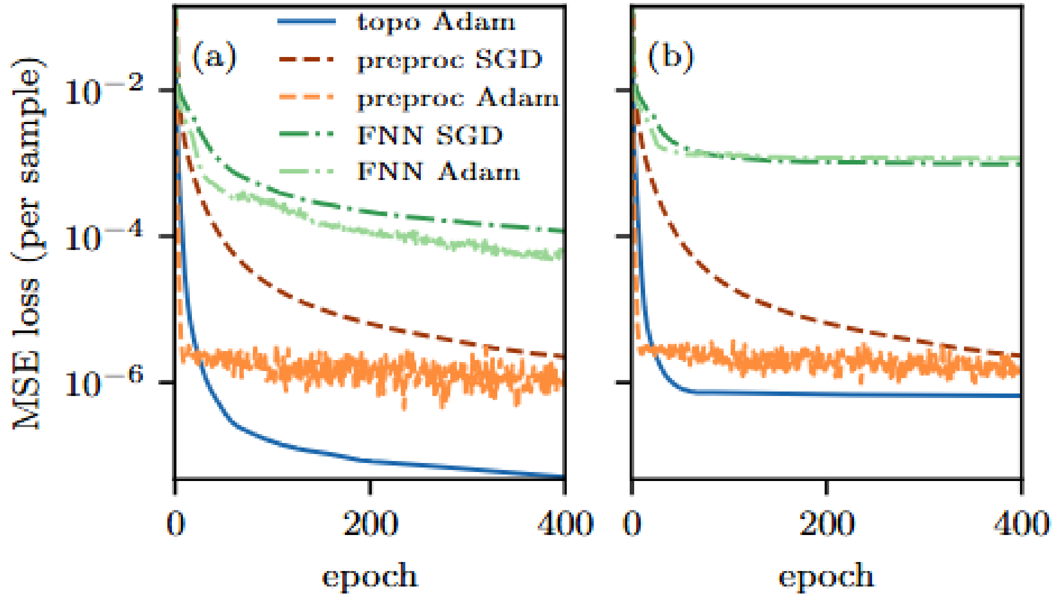
Dataset – SSH and ESSH models,
160k samples: wave functions -> winding number

Network with proper first layer and many other layers – "preprocessed"



"topo" network – follows exact structure of RSWN formula

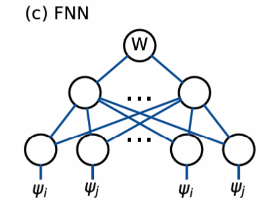




Training loss – 4 order of magnitude better!

Why regular neural networks are so “bad”?

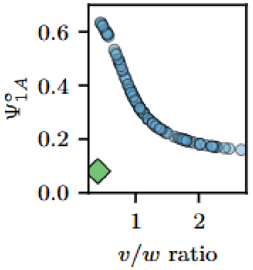
- Selected powers

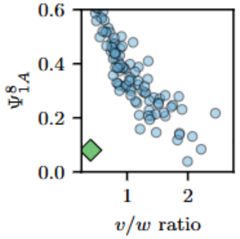


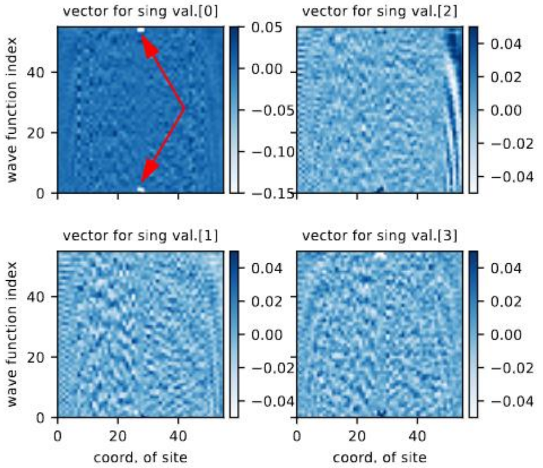
$$Out_i = ReLu\ or\ Tahn\ or\ Sigmoid\ (W * Input_i) =$$
$$= ax + bx^2 + cx^3 + dx^4 + \cdots.$$

- “Single-neuron networks”
- Ψ_x^n

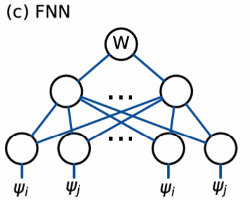
W= $\Theta(100(\Psi - 0.4))$

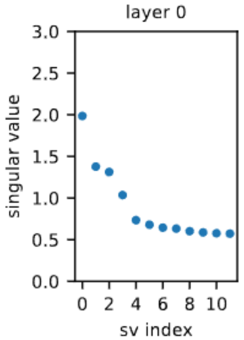






- Tendency to count edge states

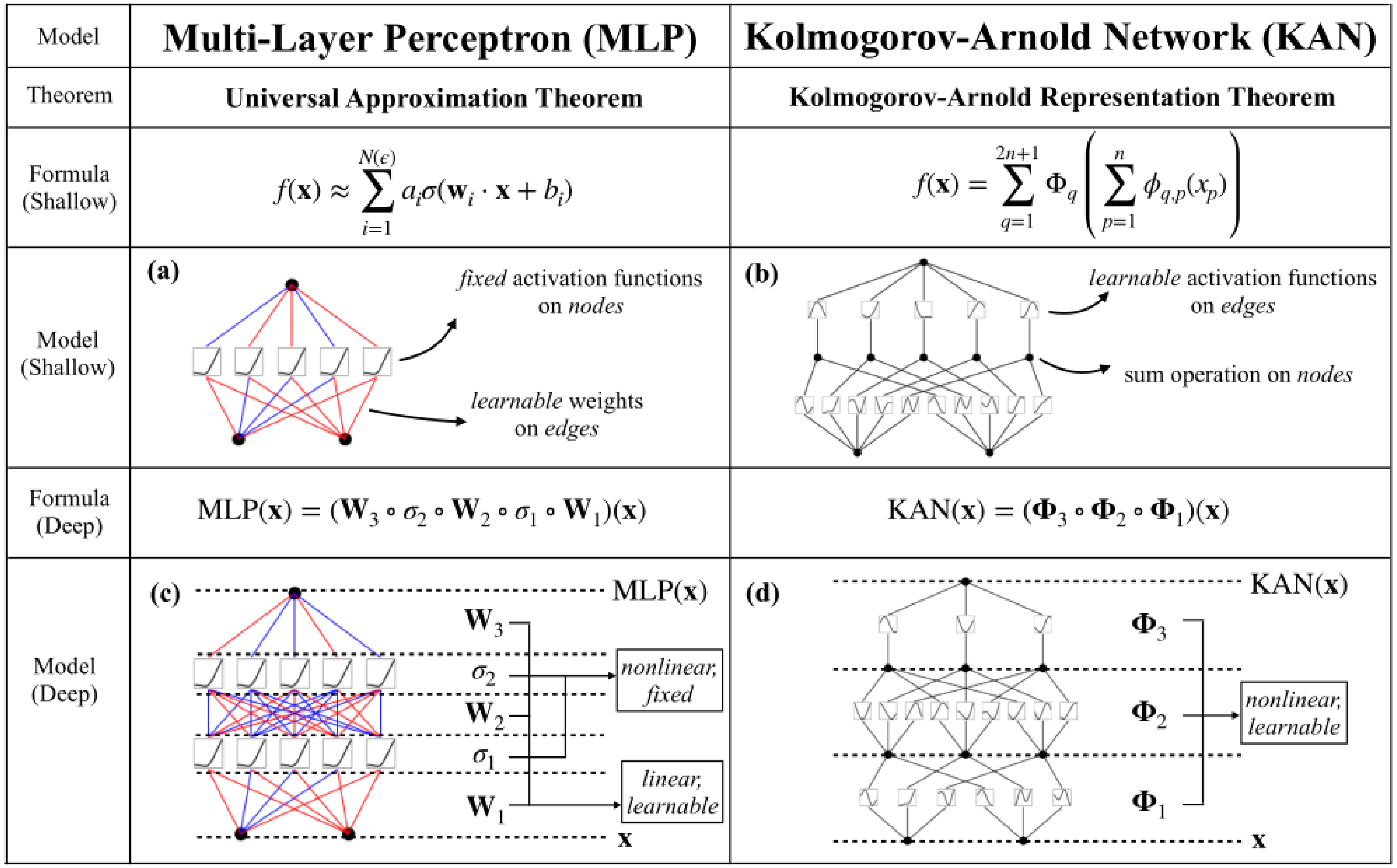




P. Zhang, H. Shen, and H. Zhai, Machine learning topological invariants with neural networks, PRL 120, 066401 (2018).
M. D. Caio, M. Caccin, P. Baireuther, T. Hyart, M. Fruchart, Machine learning assisted measurement of local topological invariants, arXiv:1901.03346 (2019)
Y. Zhangб E.-A. Kim, Quantum loop topography for machine learning, PRL 118, 216401 (2017)
N. Sun, J. Yi, P. Zhang, H. Shen, and H. Zhai, Deep learning topological invariants of band insulators, PRB 98, 085402 (2018).
P. Baireuther, M. Plodzie´n, T. Ojanen, J. Tworzydlo, T. Hyart, Identifying chern numbers of superconductors from local measurements, SciPost Phys. Core 6, 087 (2023)

Side knowledge: Kolmogorov-Arnold networks

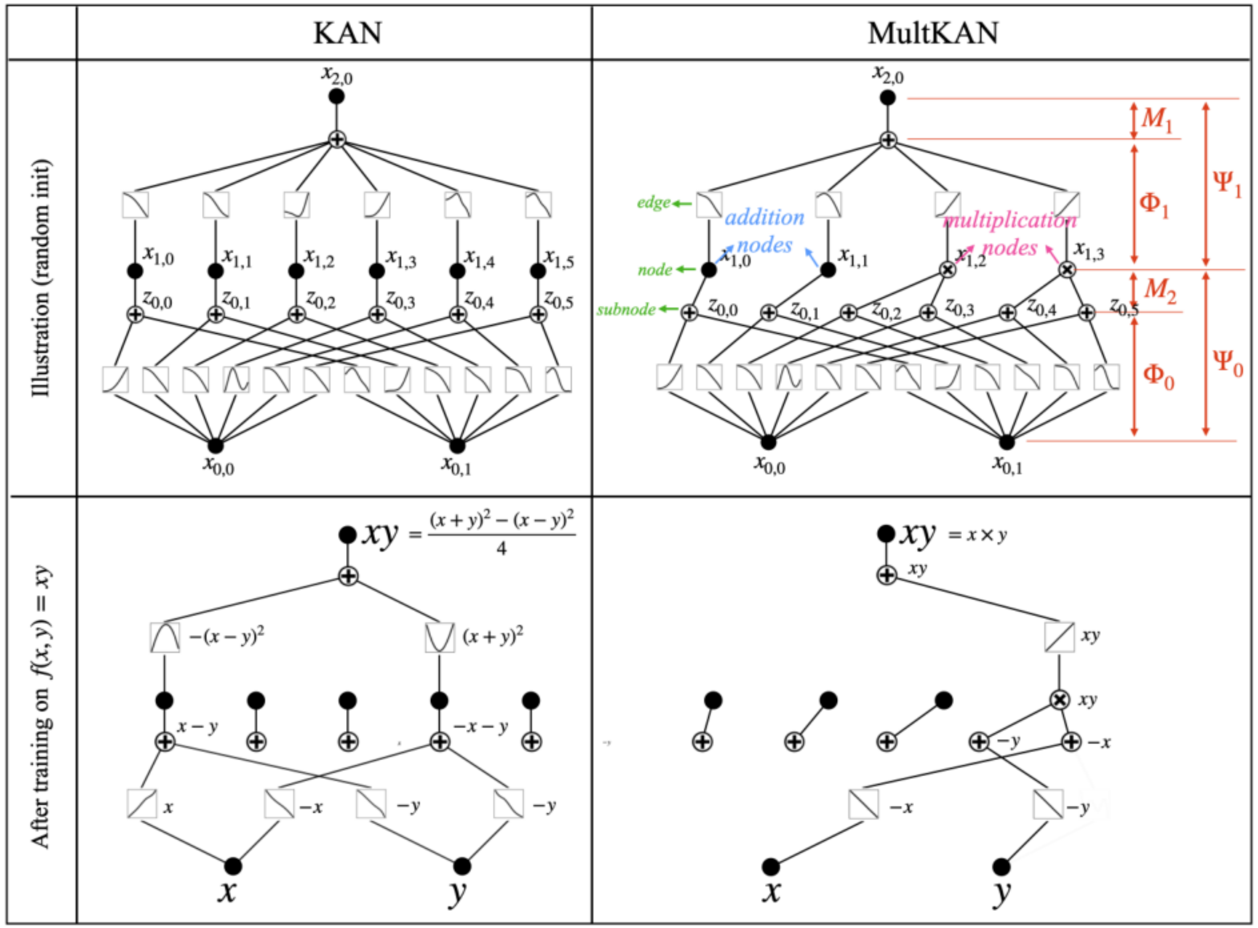
Arxiv:2404.19756, Max Tegmark group

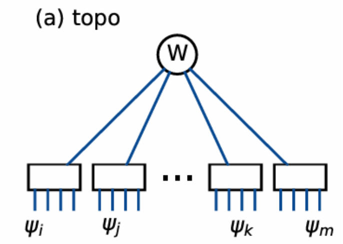


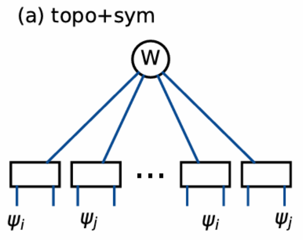
MLP – as hammer

KAN – as violine

KAN 2.0 We have KAN 2.0 for topo insulators, “violin”







- Multiplying layer in the main ingredient

- Possible to learn /

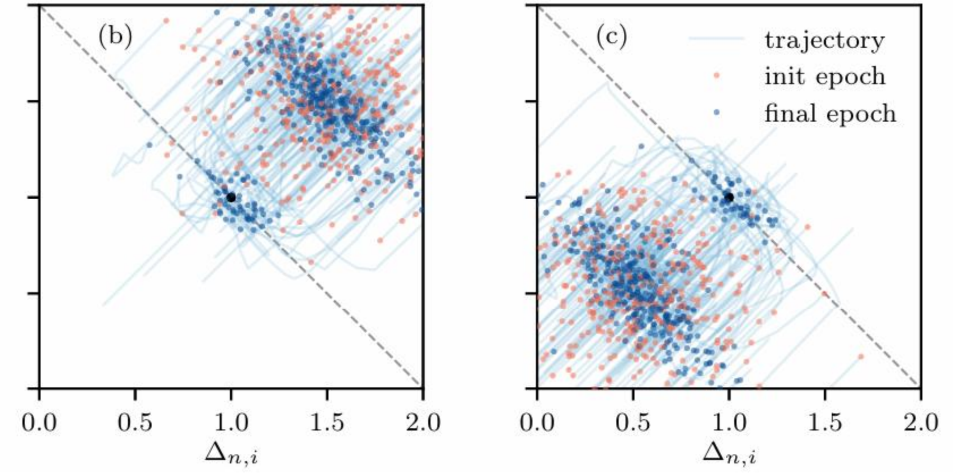
Or guess from problem type

Tegmark group,
arXiv:2408.10205

Opening black box: weights of trained network

- If physical model (as SSH) contains symmetry – many possible “exact” minima for NN
- Trained weights of 1-layer “topo” NN – satisfy inversion symmetry $\Psi_n(x) = (-1)^n \Psi_n(-x) \ \rightarrow$ $\Delta_{n,x} + \Delta_{n,-x} = 2$

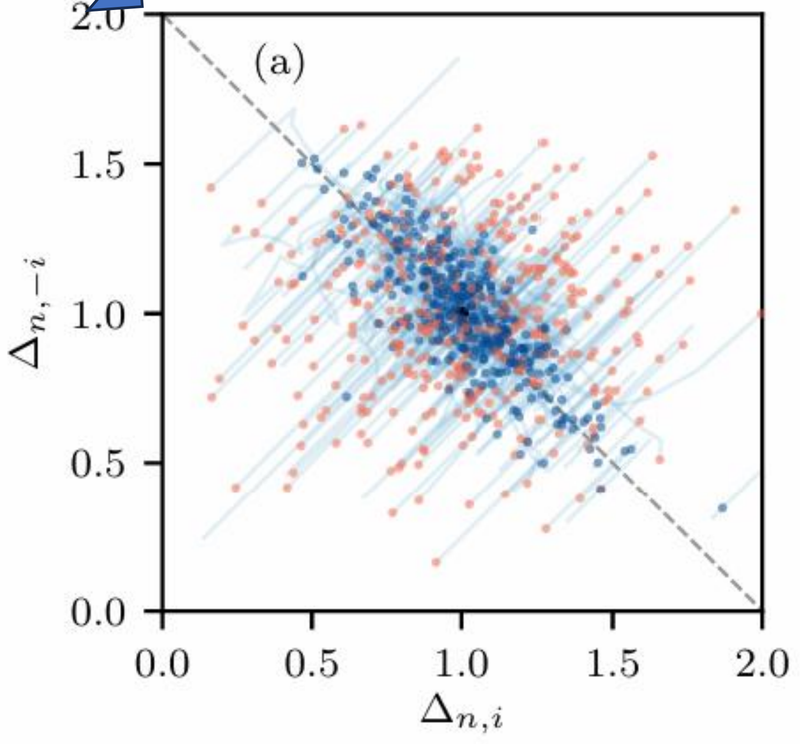
- Architecture of “topo” NN requires proper **tensor network input**



Exact formula to learn

Orange color – initial weights of NN

Dark color – later training epoch



Outlook

- A road towards exploring complicated topo insulators (amorphous? …)

- next target: identifying topological invariant by collection of tensor-neural networks

- Neural network could be a “numerical topological invariant”

QMAI group and collaborators

Juan Torres Luna

Guliuxin Jin

Lukas Splitthoff

Sibren van der Meer

Stan Bergkamp

Badr Zouggari

Naoual El Yazidi

Eliška Greplová

Next talk

Physics informed neural networks for topological invariants

Motivation:

- Machine-assisted classification of phase in experiment
- Part of states / measurables not available
- Disorder
- Make computational analogue of topo invariant