MLPhYs

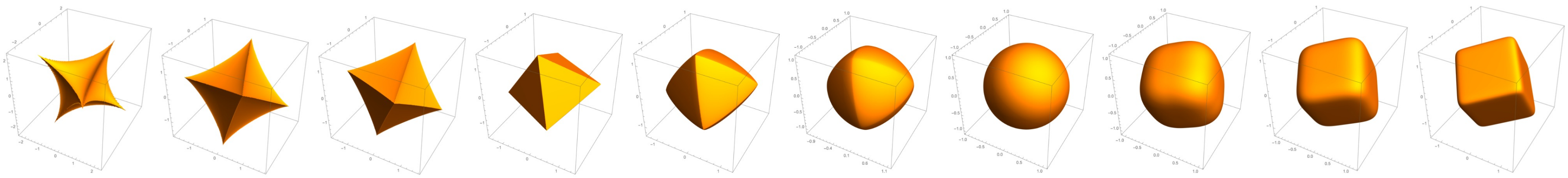
Foundation of "Machine Learning Physics"

学習物理学の創成

Grant-in-Aid for Transformative Research Areas (A)

Neural Polytopes

Based on: 1) "Neural polytopes" (ArXiv:2307.00721 [cs.LG]) accepted at ICML2023 workshop poster
2) "Multi-body wave function of ground and low-lying excited states using unornamented deep neural network" (ArXiv:2302.08965) Phys. Rev. Research 5 (2023) 033189



Koji Hashimoto (Kyoto U., phys. dept.)
Tomoya Naito (RIKEN iTHEMS)
Hisashi Naito (Nagoya U., math dept.)

ML journey from QM to polytopes

1. Intro: NN quantum states — 4 pages
2. Solving QM with DNN — 5 pages
   ArXiv:2302.08965 [physics.comp-ph]
3. Multi-particle / interaction — 9 pages
   ArXiv:2302.08965 [physics.comp-ph]
4. Discrete geometry — 5 pages
   Unpublished (on-going work)
5. Neural polytopes — 6 pages
   ArXiv:2307.00721 [cs.LG]

① Neural Network Quantum States

Find ground state wave function $\psi(s_1, s_2, \cdots, s_N)$

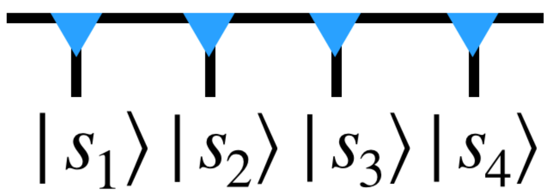
Q : Minimize its energy *E* for a given Hamiltonian *H*,

$$E = \frac{\sum_{s_1,\cdots s_N, s'_1,\cdots,s'_N} \psi^\dagger(s'_1,\cdots,s'_N)\hat{H}_{s'_1,\cdots s'_N, s_1,\cdots,s_N}\psi(s_1,\cdots,s_N)}{\sum_{s_1,\cdots s_N}\psi^\dagger(s_1,\cdots,s_N)\psi(s_1,\cdots,s_N)}$$

A : Use ansatz and optimize parameters!

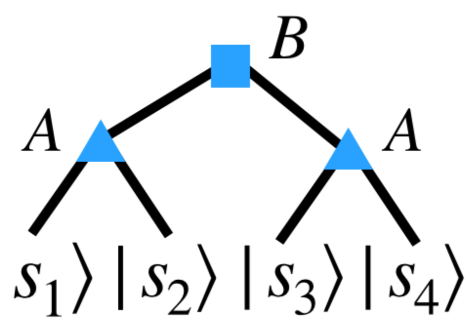
- Matrix product states

$$\psi(s_1, s_2, \cdots) = \mathrm{tr}[A^{(s_1)}A^{(s_2)}\cdots]$$



- Tensor network states

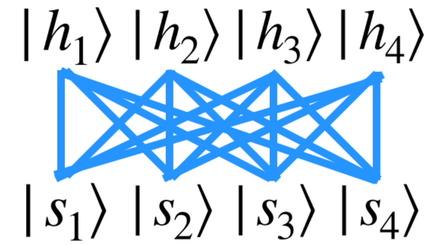
$$\psi(s_1, s_2, \cdots) = \sum_{m,n} B_{mn}A_{ms_1s_2}A_{ns_3s_4}$$



1. Neural Network Quantum States

Neural network can be wave functions

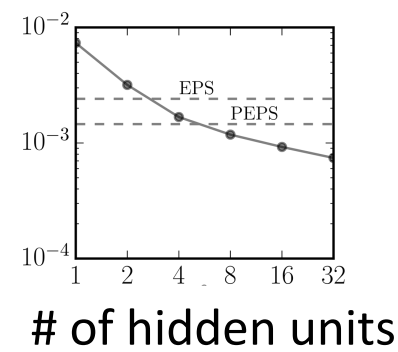
- Boltzmann machine states [Carleo, Troyer `17], [Nomura, Darmawan, Yamaji, Imada `17], ..



$$\psi(s_1, \cdots, s_N) = \sum_{h_A} \exp\left[\sum_a a_a s_a + \sum_A b_A h_A + \sum_{a,A} J_{aA} s_a h_A\right]$$

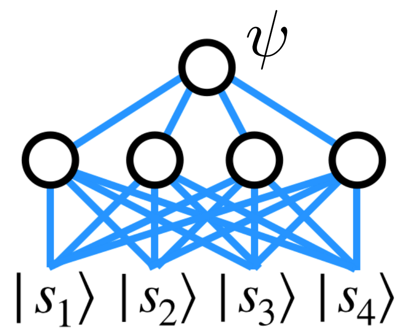
Ex) 2-d antiferromagnetic Heisenberg model was better-approximated

Energy with RBM states



- Feedfoward network states [Saito `18], ..

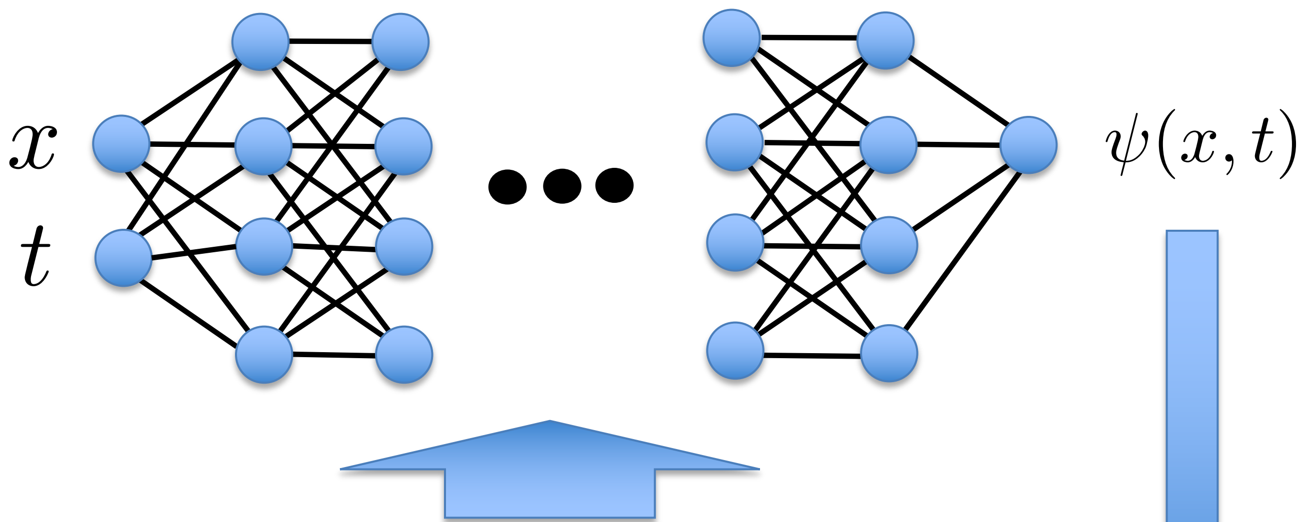
$$\psi(s_1, \cdots, s_N) = \sum_i f_i \, \sigma\left(\sum_j W_{ij} s_j + b_i\right)$$



① Neural Network Quantum States

PINN (Physics-informed neural networks)

[Raissi, Perdikaris, Karniadakis `17], ..



Loss $\mathcal{L} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{EoM}} + \mathcal{L}_{\text{BC}}$

- Physics observations $\psi(x = x_n, t = t_i)] = \psi_{\text{data}}$
- Equations of motion $\mathcal{F}[\psi(x,t)] = 0$
- Boundary conditions $\mathcal{B}[\psi(x = x_0, t)] = 0$

① Neural Network Quantum States 4/4

My motivation : continuum ⇔ discrete

So far, most of the work are separated to either “discrete” inputs or “continuous” inputs.

- Discretization effects in DNN approximation of continuous systems is small enough?
- Concepts in continuous systems – such as topology – will be modified or surviving?
- How the wave function “space” is generated by DNN and machine learning?

ML journey from QM to polytopes

1. Intro: NN quantum states — 4 pages
2. Solving QM with DNN — 5 pages
   ArXiv:2302.08965 [physics.comp-ph]
3. Multi-particle / interaction — 9 pages
   ArXiv:2302.08965 [physics.comp-ph]
4. Discrete geometry — 5 pages
   Unpublished (on-going work)
5. Neural polytopes — 6 pages
   ArXiv:2307.00721 [cs.LG]

② Solving QM with DNN

QM on a lattice, suitable for DNN

Question

For a given Hamiltonian, solve for ground and excited states.
1d or multi-d. 1-particle or many particle. Boson or fermion.

Strategy

1) Prepare your QM Hamiltonian on a lattice.

2) DNN input : spatial coordinate values (x,y,..) on a lattice
DNN output : wave function value f(x,y,..)
loss function : Energy expectation value, for the whole lattice

3) Train DNN and obtain the ground state wave function.

② Solving QM with DNN

1d harmonic oscillator on a lattice

Strategy 1) Prepare your QM Hamiltonian on a lattice.

$$H=-\frac{\hbar^2}{2m}\sum_j \Delta_j+\sum_j V^{\text{ext}}(\boldsymbol{r}_j)+\frac{1}{2}\sum_{j\neq k}V^{\text{int}}(\boldsymbol{r}_j,\boldsymbol{r}_k)$$

$$\simeq \tilde{H}=-\frac{1}{2h^2}\tilde{T}+\tilde{V}^{\text{ext}}$$

$$\tilde{T}=\begin{pmatrix}-2 & 1 & 0 & \dots & 0 & 0 & 0\\ 1 & -2 & 1 & \dots & 0 & 0 & 0\\ 0 & 1 & -2 & \dots & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & \dots & -2 & 1 & 0\\ 0 & 0 & 0 & \dots & 1 & -2 & 1\\ 0 & 0 & 0 & \dots & 0 & 1 & -2\end{pmatrix}\qquad \tilde{V}^{\text{ext}}=\begin{pmatrix}V_1^{\text{ext}} & 0 & 0 & \dots & 0 & 0 & 0\\ 0 & V_2^{\text{ext}} & 0 & \dots & 0 & 0 & 0\\ 0 & 0 & V_3^{\text{ext}} & \dots & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & \dots & V_{M-3}^{\text{ext}} & 0 & 0\\ 0 & 0 & 0 & \dots & 0 & V_{M-2}^{\text{ext}} & 0\\ 0 & 0 & 0 & \dots & 0 & 0 & V_{M-1}^{\text{ext}}\end{pmatrix}$$

$$\psi\simeq\tilde{\psi}=\begin{pmatrix}\psi_1\\ \psi_2\\ \psi_3\\ \vdots\\ \psi_{M-3}\\ \psi_{M-2}\\ \psi_{M-1}\end{pmatrix}$$

$$V_j^{\text{ext}}=V^{\text{ext}}(x_j),\ \psi_j=\psi(x_j),\ x_j=-x_{\max}+hj,$$

$$h\sqrt{\textstyle\sum_j\tilde{\psi}_j^2}=1,\quad h=2x_{\max}/M$$

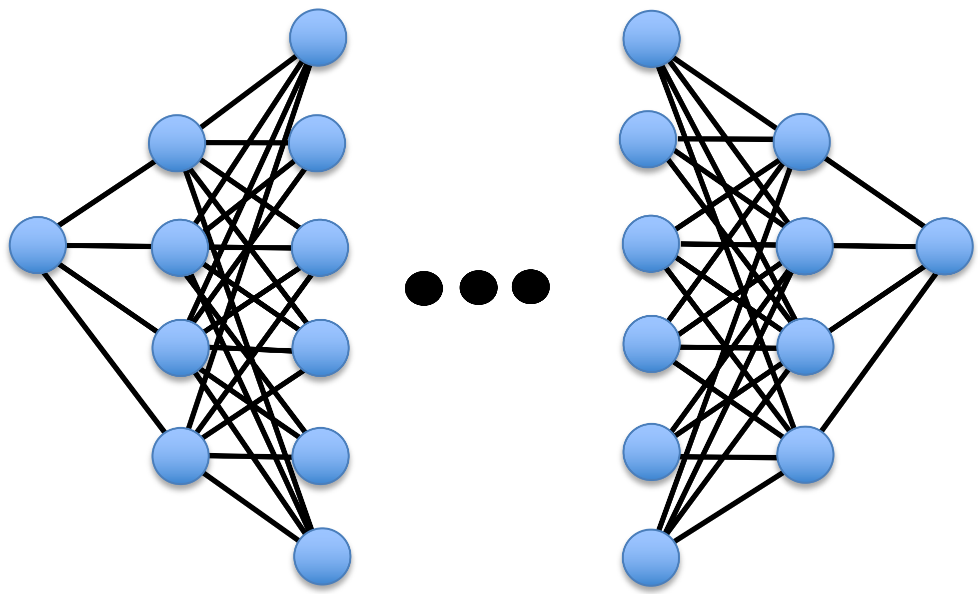
② Solving QM with DNN

1 → 1 DNN is prepared

<u>Strategy</u>

2) DNN input : spatial coordinate values (x,y,..) on a lattice
DNN output : wave function value f(x,y,..)
loss function : Energy expectation value, for the whole lattice

$x_j = -x_{\max} + hj,$



$\psi_j = \psi(x_j)$

Activation function: $\mathrm{softplus}(x) = \log(1 + e^x)$

② Solving QM with DNN

Successful training with good accuracy

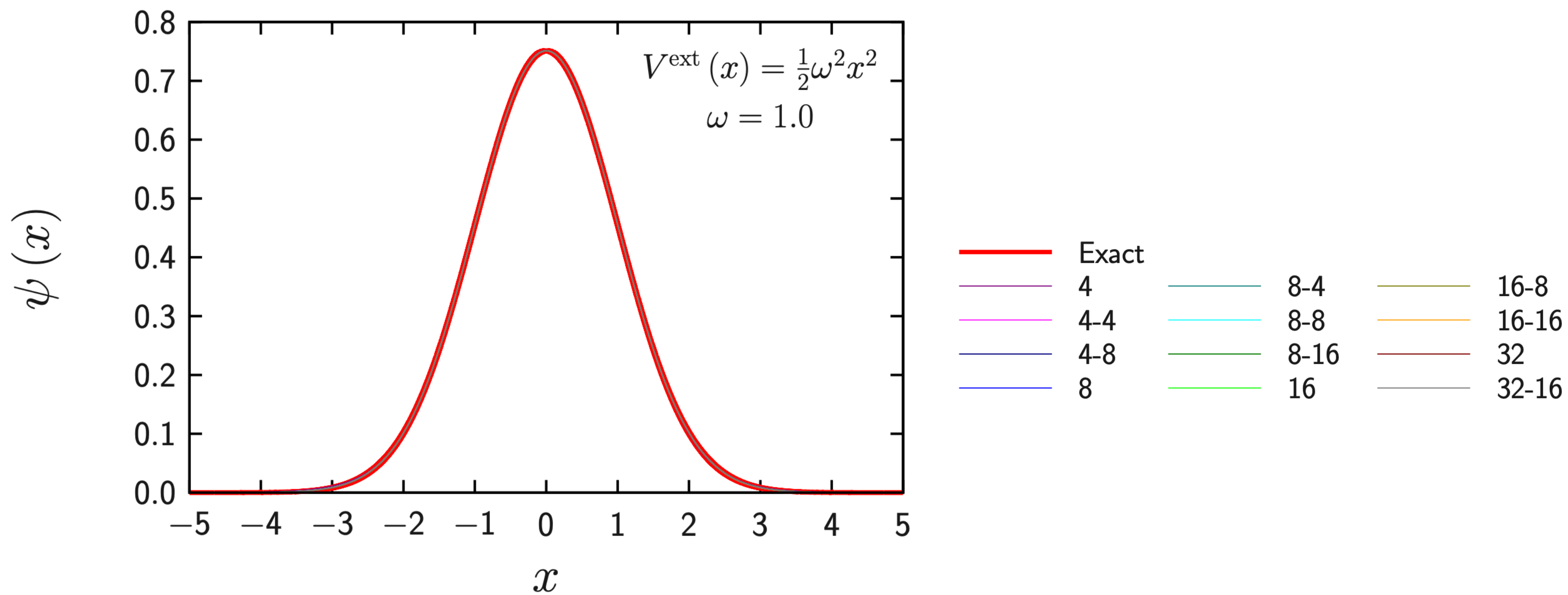
<u>Strategy</u> 3) Train DNN and obtain the ground state wave function.

$$V^{\text{ext}}(x) = \frac{1}{2}\omega^2 x^2$$

ω	# of Unit		Energy		
	1st Layer	2nd Layer	Kinetic	Potential	Total
1.0	4	—	+0.250043	+0.250032	+0.500075
1.0	4	4	+0.250006	+0.249996	+0.500002
1.0	4	8	+0.250001	+0.249997	+0.499998
1.0	8	—	+0.250002	+0.250002	+0.500004
1.0	8	4	+0.250000	+0.249998	+0.499998
1.0	8	8	+0.250004	+0.249996	+0.500001
1.0	8	16	+0.249999	+0.249999	+0.499997
1.0	16	—	+0.250000	+0.249999	+0.499999
1.0	16	8	+0.250000	+0.249998	+0.499998
1.0	16	16	+0.249999	+0.249998	+0.499997
1.0	32	—	+0.250000	+0.249999	+0.499998
1.0	32	16	+0.249999	+0.249999	+0.499998

② Solving QM with DNN

DNN wave function matches



$$\psi_{\mathrm{gs}}(x) = \frac{1}{3.7451}\,\mathrm{softplus}\left(a_{\mathrm{gs}}(x)\right),$$
$$a_{\mathrm{gs}}(x) = 2.4069a_1(x) - 1.8344a_2(x) - 1.9778a_3(x) + 2.3484a_4(x) - 4.8998,$$
$$a_1(x) = \mathrm{softplus}\left(0.35953x + 3.9226\right),$$
$$a_2(x) = \mathrm{softplus}\left(2.5821x + 0.033213\right),$$
$$a_3(x) = \mathrm{softplus}\left(-0.65170x + 2.9574\right),$$
$$a_4(x) = \mathrm{softplus}\left(0.15421x + 2.2016\right)$$

ML journey from QM to polytopes

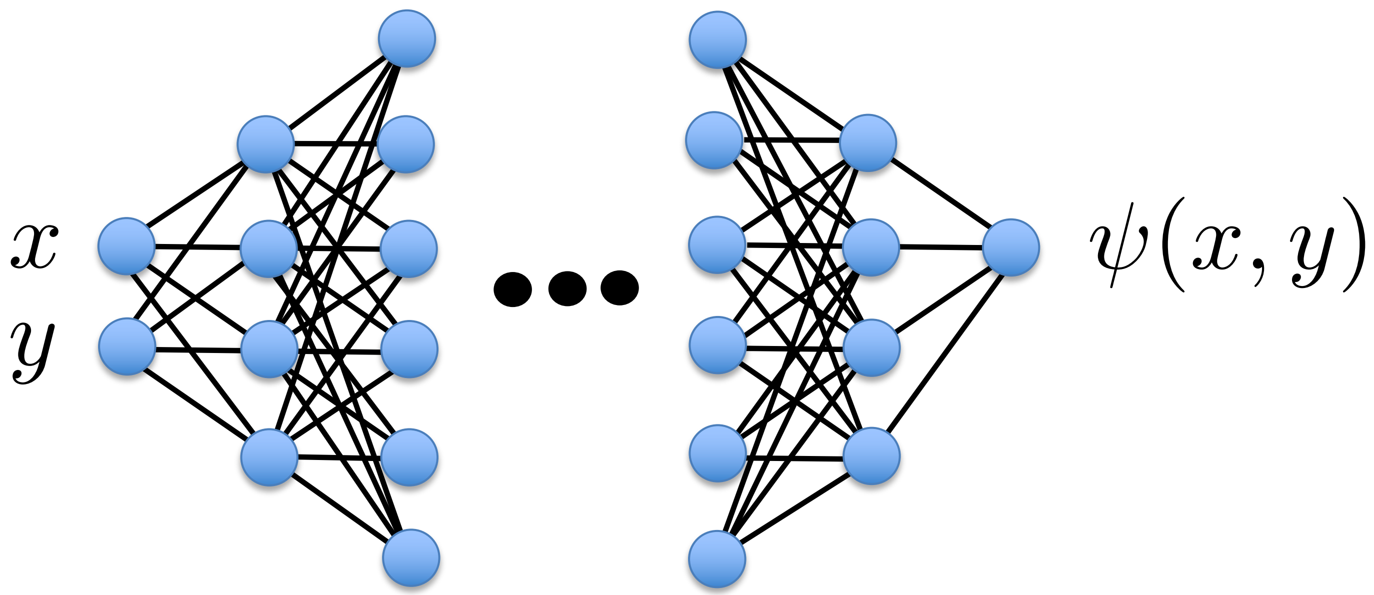
1. Intro: NN quantum states — 4 pages
2. Solving QM with DNN — 5 pages
   ArXiv:2302.08965 [physics.comp-ph]
3. Multi-particle / interaction — 9 pages
   ArXiv:2302.08965 [physics.comp-ph]
4. Discrete geometry — 5 pages
   Unpublished (on-going work)
5. Neural polytopes — 6 pages
   ArXiv:2307.00721 [cs.LG]

③ Multi-particle / interaction

2 particles, 1d harmonic oscillator

(x,y) : location of the first and second particle

Introduction of interaction $V^{\mathrm{int}}(x,y) = \lambda \exp\left(-|x-y|\right)$



Simplicity: Symmetrization (boson) is imposed afterwards!!

③ Multi-particle / interaction

2 particles, 1d harmonic oscillator

Particles	ω	λ	Energy
Boson	1.0	−1.00	−89.869381
Boson	1.0	−0.25	−19.848949
Boson	1.0	+0.00	+0.999927
Boson	1.0	+0.25	+3.298725
Boson	1.0	+1.00	+3.835173
Boson	5.0	−1.00	−87.554311
Boson	5.0	−0.25	−17.203647
Boson	5.0	+0.00	+4.997829
Boson	5.0	+0.25	+21.149827
Boson	5.0	+1.00	+31.804917
Boson	10.0	−1.00	−84.129658
Boson	10.0	−0.25	−13.118213
Boson	10.0	+0.00	+9.991009
Boson	10.0	+0.25	+32.287424
Boson	10.0	+1.00	+72.688350
Fermion	1.0	−1.00	−71.409493
Fermion	1.0	−0.25	−11.369632
Fermion	1.0	+0.00	+1.999931
Fermion	1.0	+0.25	+3.298786
Fermion	1.0	+1.00	+3.839178
Fermion	5.0	−1.00	−68.409494
Fermion	5.0	−0.25	−7.207718
Fermion	5.0	+0.00	+9.995902
Fermion	5.0	+0.25	+21.385884
Fermion	5.0	+1.00	+31.804915
Fermion	10.0	−1.00	−63.026667
Fermion	10.0	−0.25	+0.302312
Fermion	10.0	+0.00	+19.975414
Fermion	10.0	+0.25	+38.060907
Fermion	10.0	+1.00	+73.245097

Compare:

For exact sol. at $\lambda = 0$

Boson : $E_{\mathrm{gs}} = \omega$

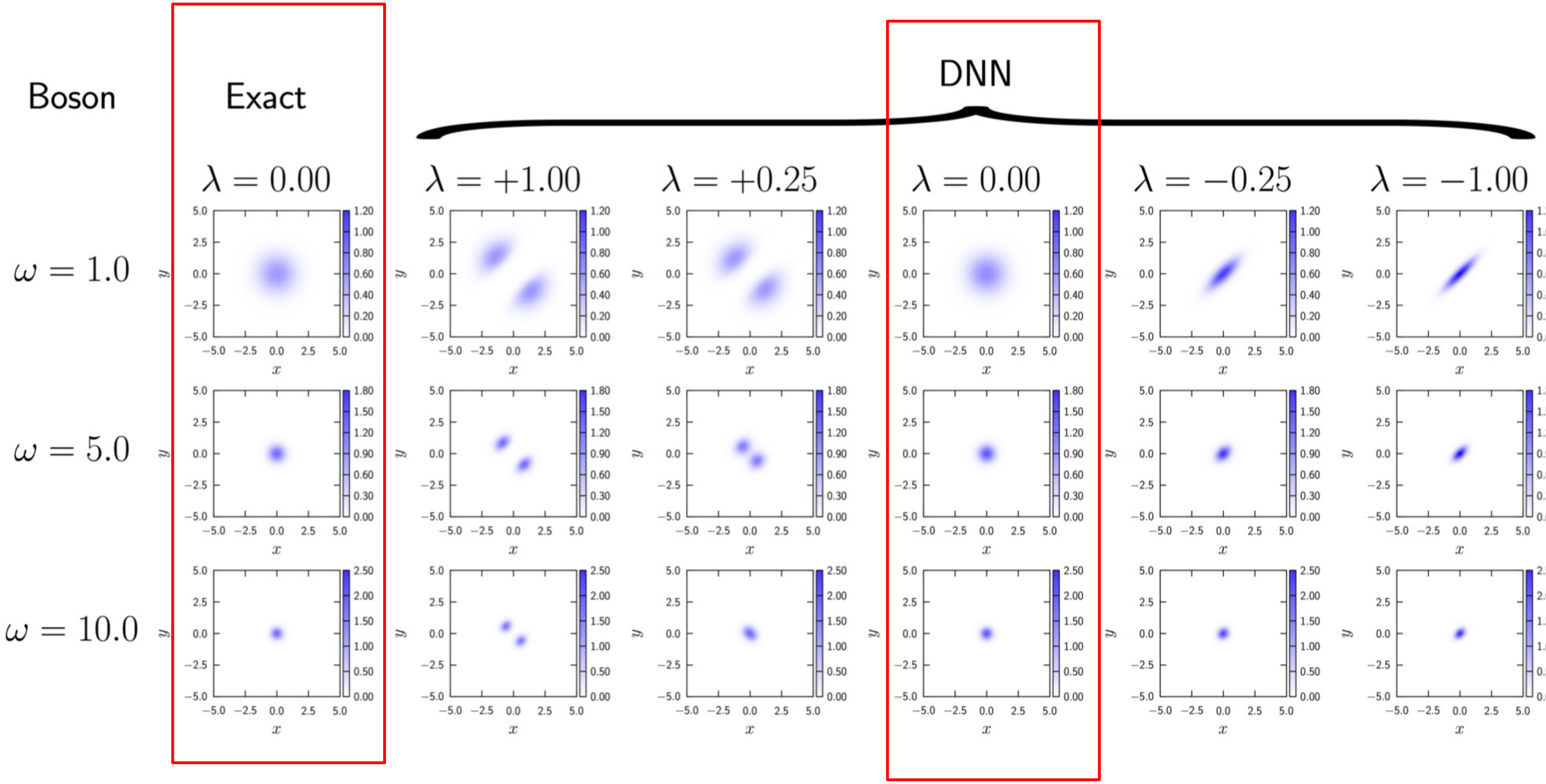
$$\psi_{\mathrm{gs}}(x, y) = \sqrt{\frac{\omega}{\pi}} \exp\left[-\frac{\omega\left(x^2 + y^2\right)}{2}\right]$$

Fermion : $E_{\mathrm{gs}} = 2\omega$

$$\psi_{\mathrm{gs}}(x, y) = \frac{\omega}{\sqrt{\pi}}(x - y) \exp\left[-\frac{\omega\left(x^2 + y^2\right)}{2}\right]$$

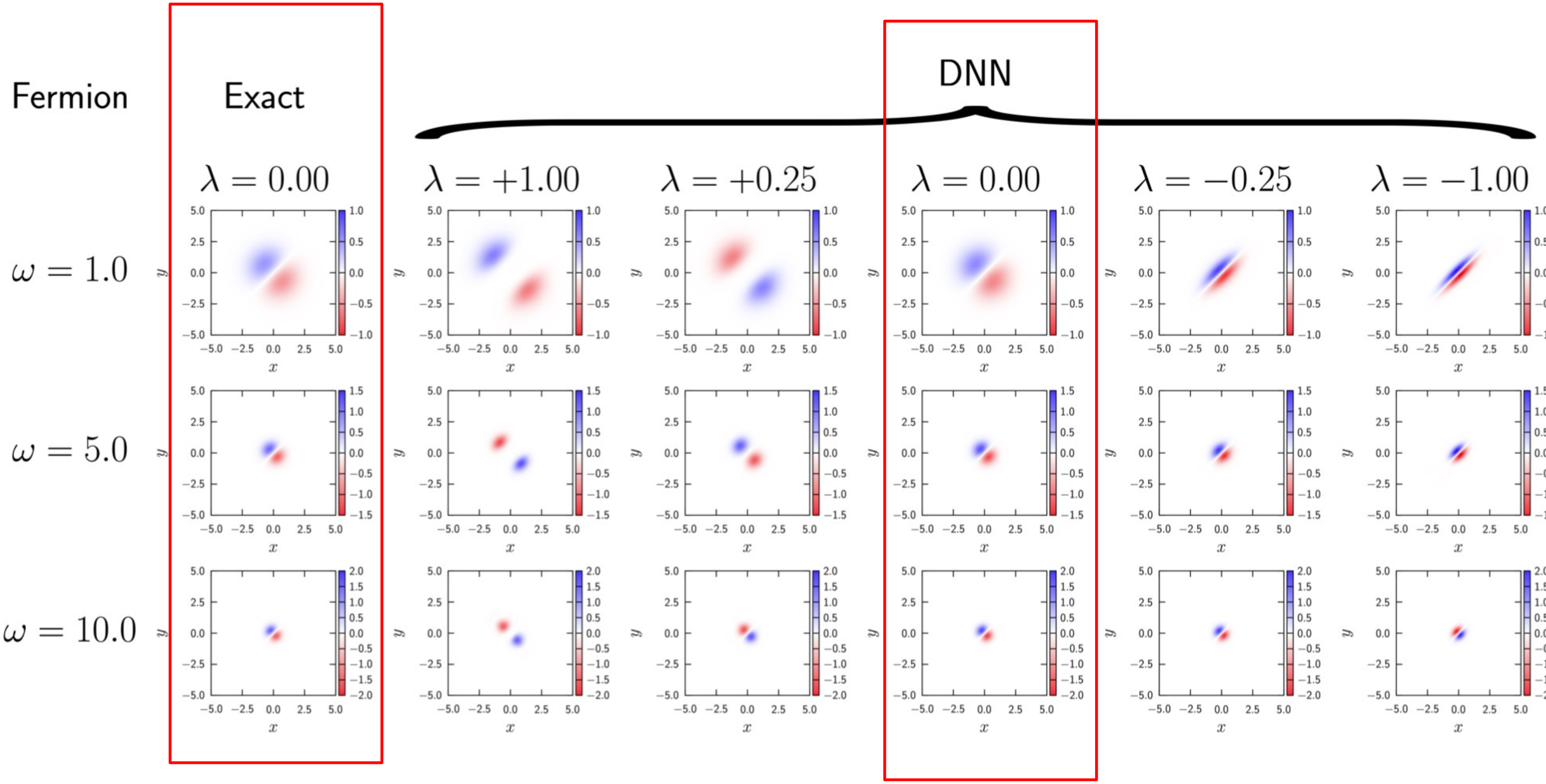
③ Multi-particle / interaction

2 particles, 1d harmonic oscillator



③ Multi-particle / interaction

2 particles, 1d harmonic oscillator

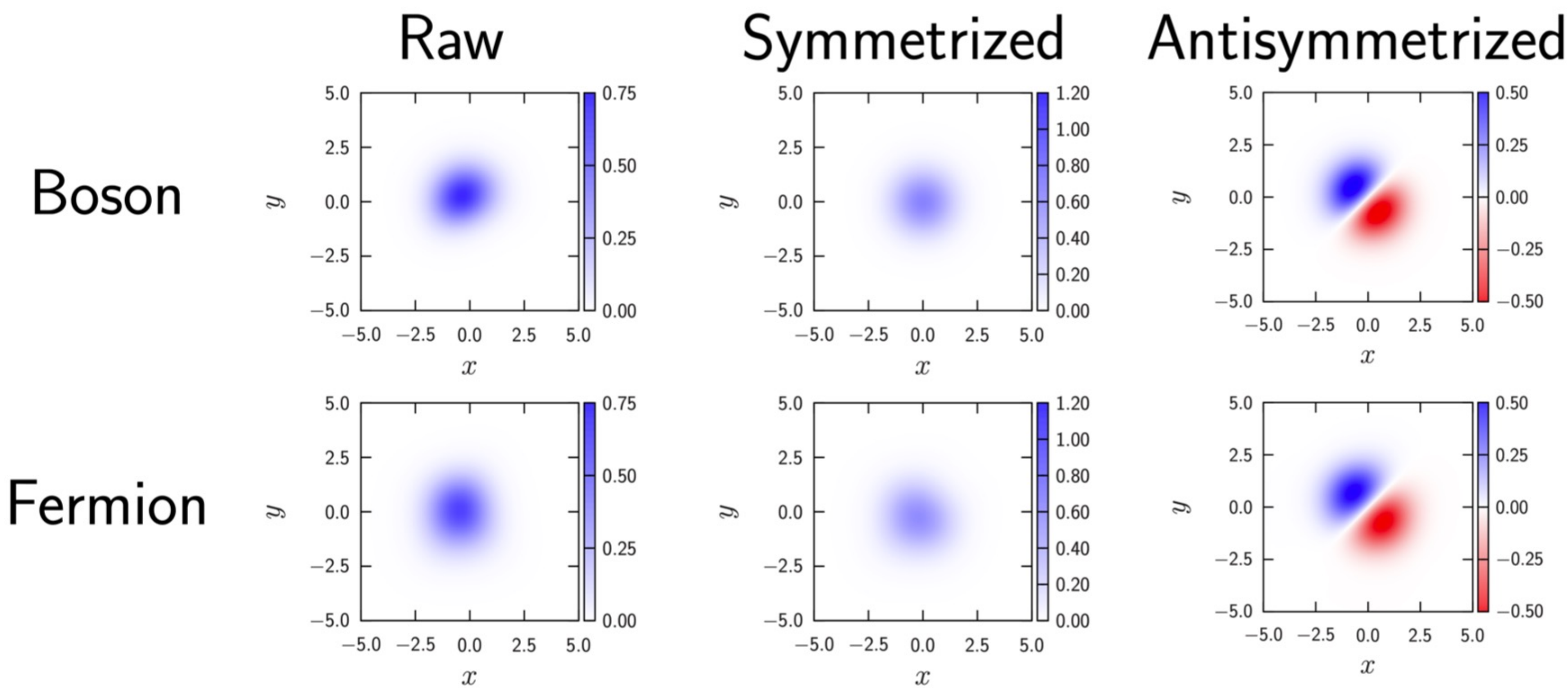


③ Multi-particle / interaction

2 particles, 1d harmonic oscillator

Generalization?!

Look inside: Raw wave function before (anti-)symmetrized



③ Multi-particle / interaction

3 particles, 1d harmonic oscillator

Training result :

$$\omega = 1.0$$
$$\lambda = 0$$

Particles	Energy
Boson	+1.497880
Fermion	+4.486830

Compare: For exact sol. at $\lambda = 0$

Boson : $E_{\text{gs}} = \frac{3}{2}\omega$

$$\psi_{\text{gs}}(x,y,z) = \left(\frac{\omega}{\pi}\right)^{3/4} \exp\left[-\frac{\omega\left(x^2+y^2+z^2\right)}{2}\right]$$

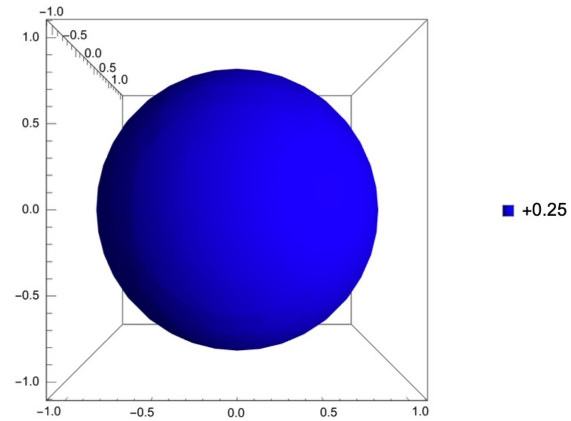
Fermion : $E_{\text{gs}} = \frac{9}{2}\omega$

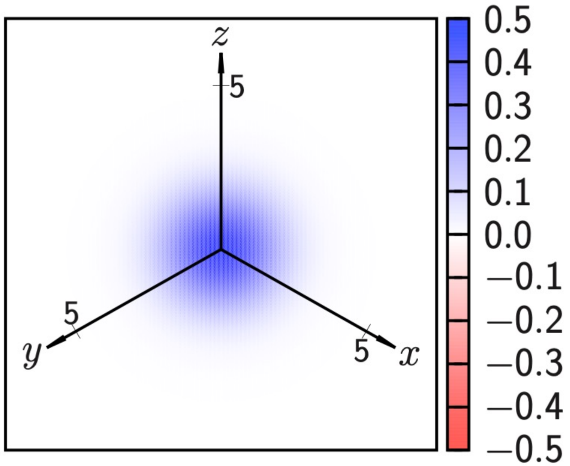
$$\psi_{\text{gs}}(x,y,z) = \left(\frac{\omega}{\pi}\right)^{3/4} \sqrt{\frac{\omega}{6}} \left[(x-y)\left(1-2\omega z^2\right) + (y-z)\left(1-2\omega x^2\right) + (z-x)\left(1-2\omega y^2\right)\right] \exp\left[-\frac{\omega\left(x^2+y^2+z^2\right)}{2}\right]$$

③ Multi-particle / interaction

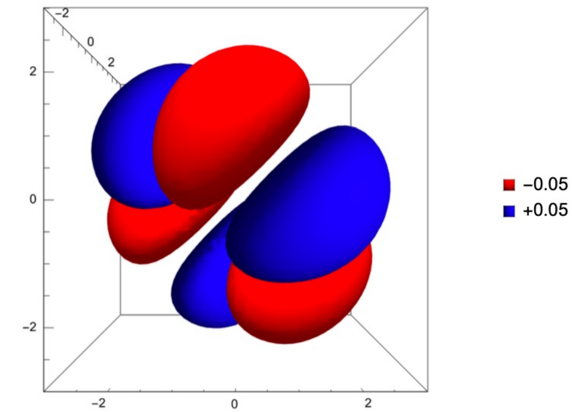
3 particles, 1d harmonic oscillator

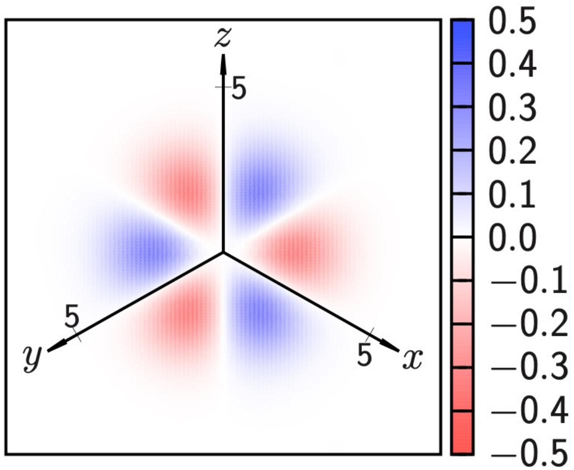
Boson :





Fermion :





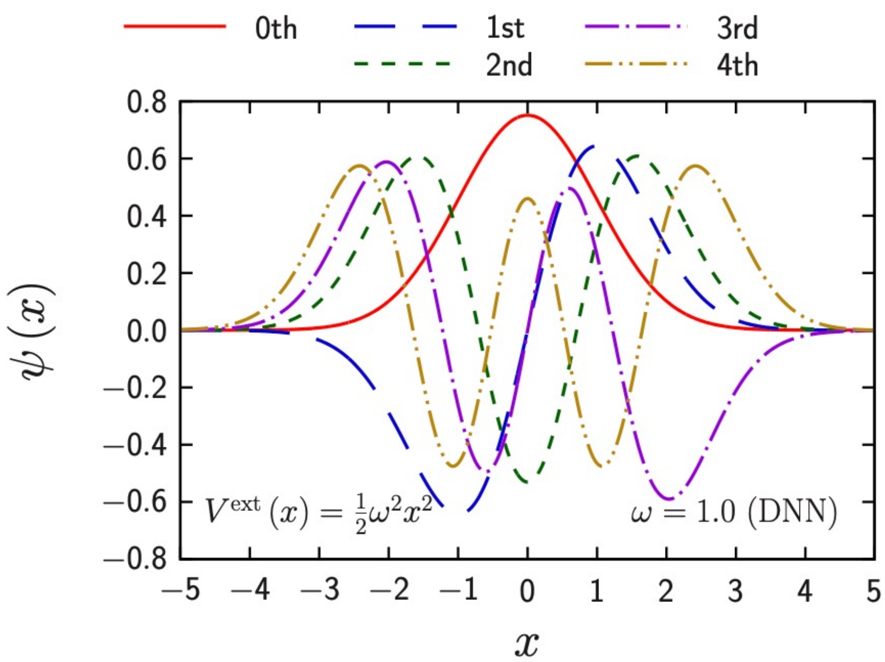
③ Multi-particle / interaction

Excited states (1 particles, 1d harmonic oscillator)

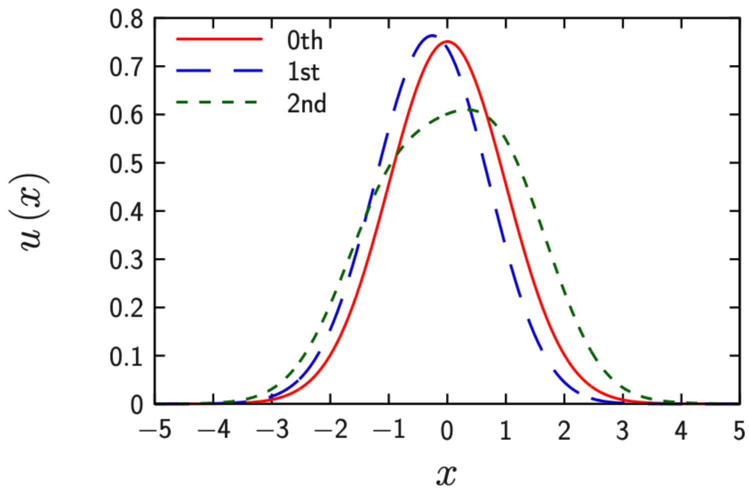
Training result :

$$\omega = 1.0$$

State	Energy
0th	+0.499998
1st	+1.499991
2nd	+2.499986
3rd	+3.500193
4th	+4.500201

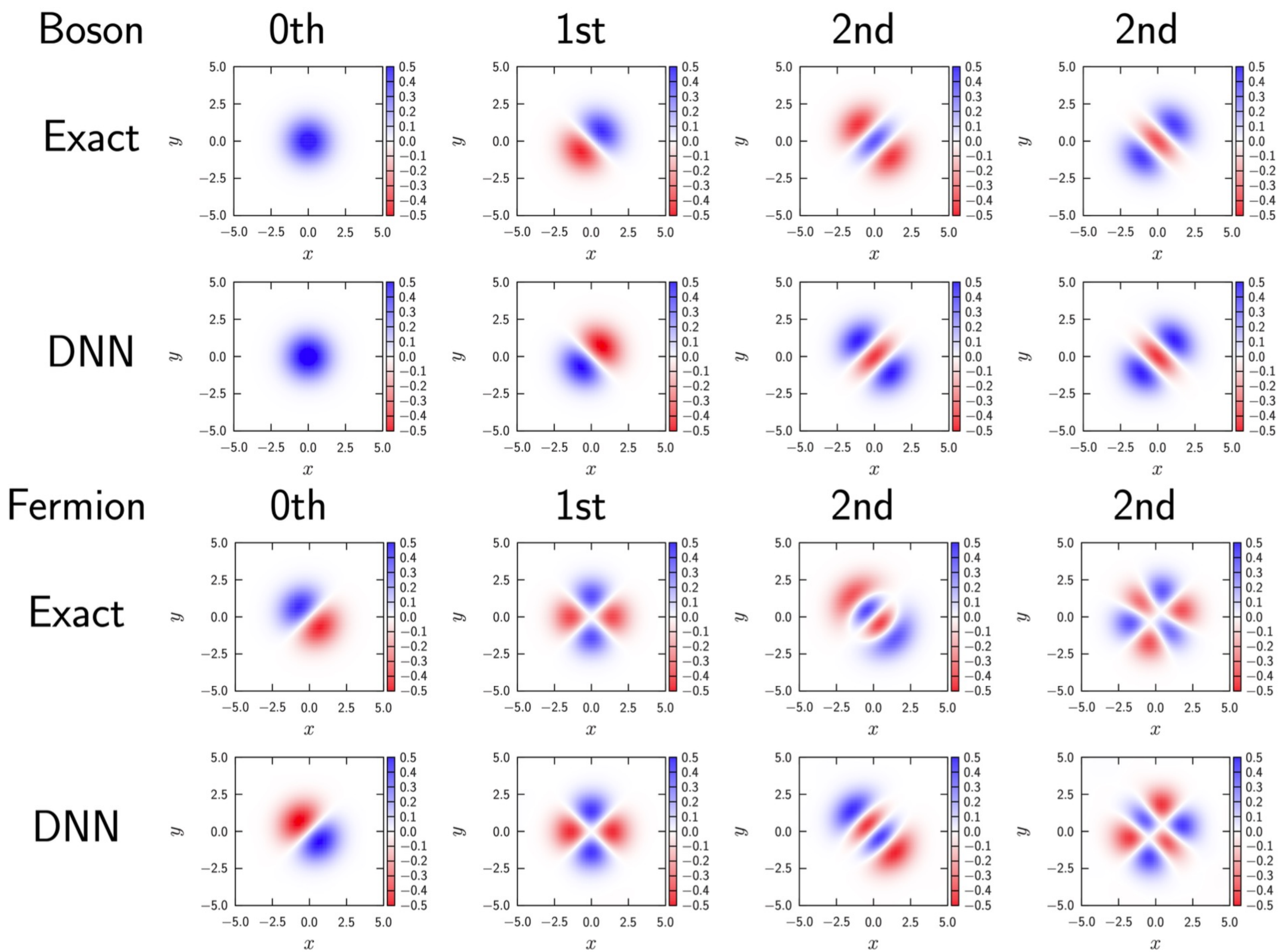


DNN output



③ Multi-particle / interaction

Excited states (2 particles, 1d harmonic oscillator)

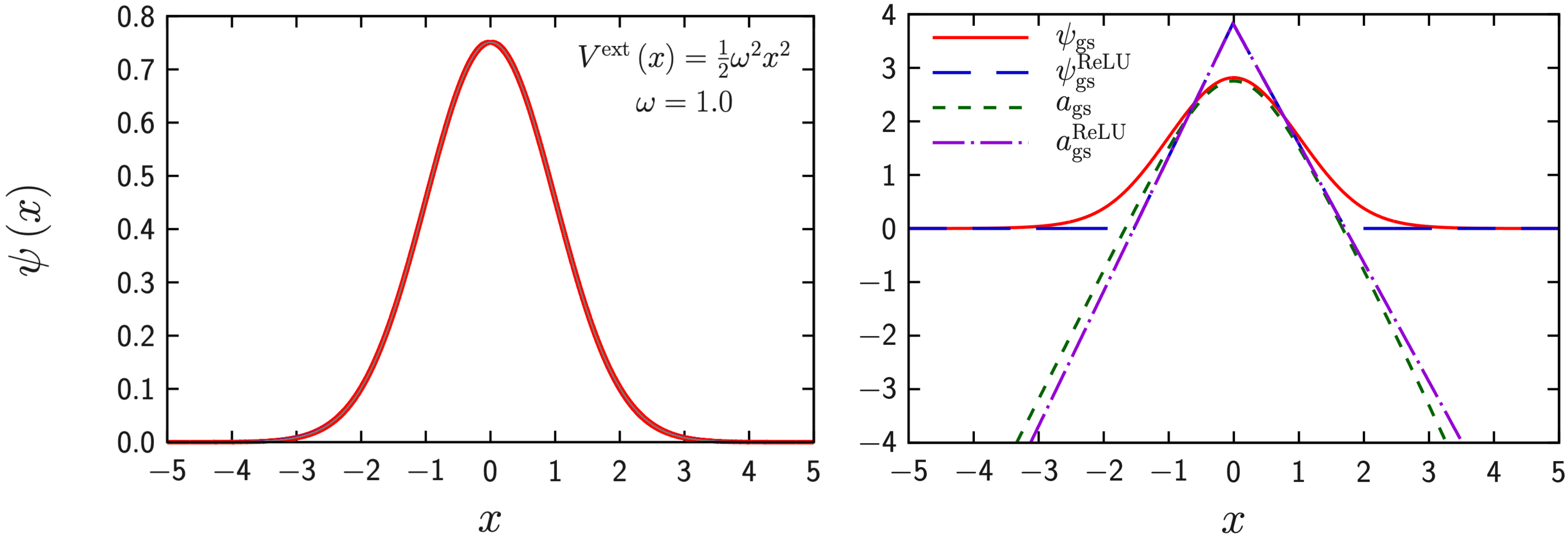


ML journey from QM to polytopes

1. Intro: NN quantum states — 4 pages
2. Solving QM with DNN — 5 pages
   ArXiv:2302.08965 [physics.comp-ph]
3. Multi-particle / interaction — 9 pages
   ArXiv:2302.08965 [physics.comp-ph]
4. Discrete geometry — 5 pages
   Unpublished (on-going work)
5. Neural polytopes — 6 pages
   ArXiv:2307.00721 [cs.LG]

④ Discrete geometry

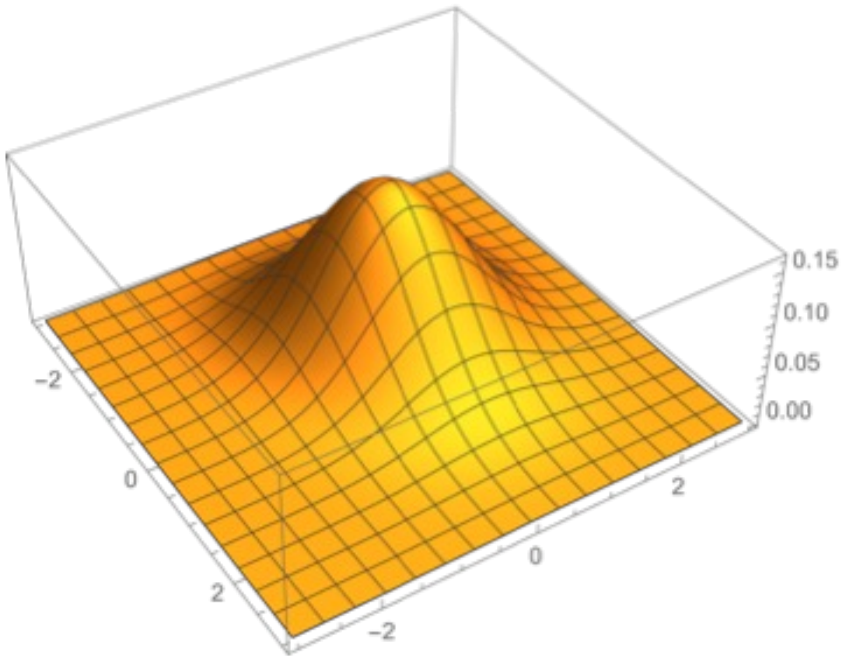
Interpretation of the DNN : discrete geometry



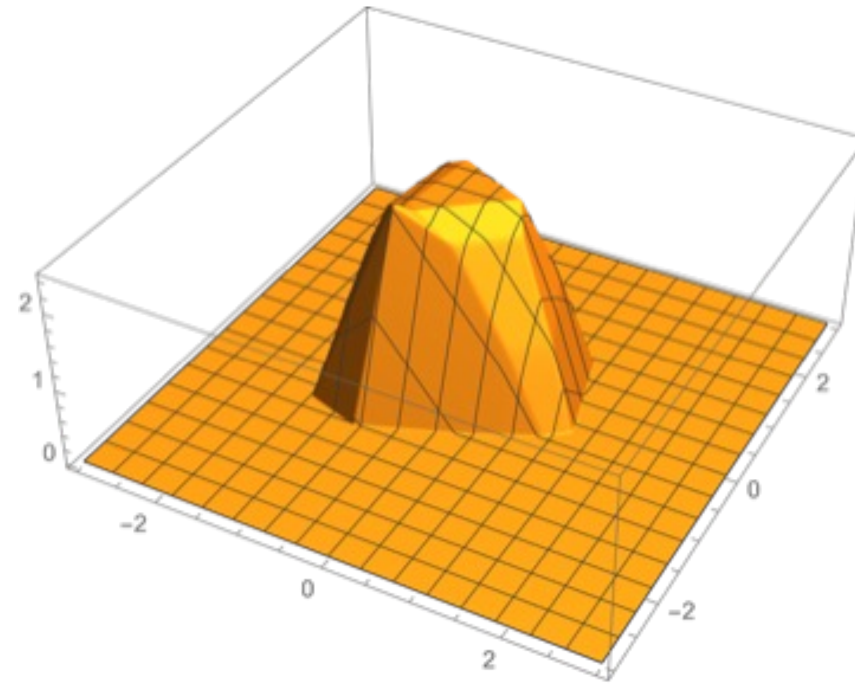
$$\psi_{\mathrm{gs}}(x) = \frac{1}{3.7451}\,\mathrm{softplus}\left(a_{\mathrm{gs}}(x)\right),$$
$$a_{\mathrm{gs}}(x) = 2.4069 a_1(x) - 1.8344 a_2(x) - 1.9778 a_3(x) + 2.3484 a_4(x) - 4.8998,$$
$$a_1(x) = \mathrm{softplus}\left(0.35953x + 3.9226\right),$$
$$a_2(x) = \mathrm{softplus}\left(2.5821x + 0.033213\right),$$
$$a_3(x) = \mathrm{softplus}\left(-0.65170x + 2.9574\right),$$
$$a_4(x) = \mathrm{softplus}\left(0.15421x + 2.2016\right),$$

④. Discrete geometry

Region-wise linear approximation



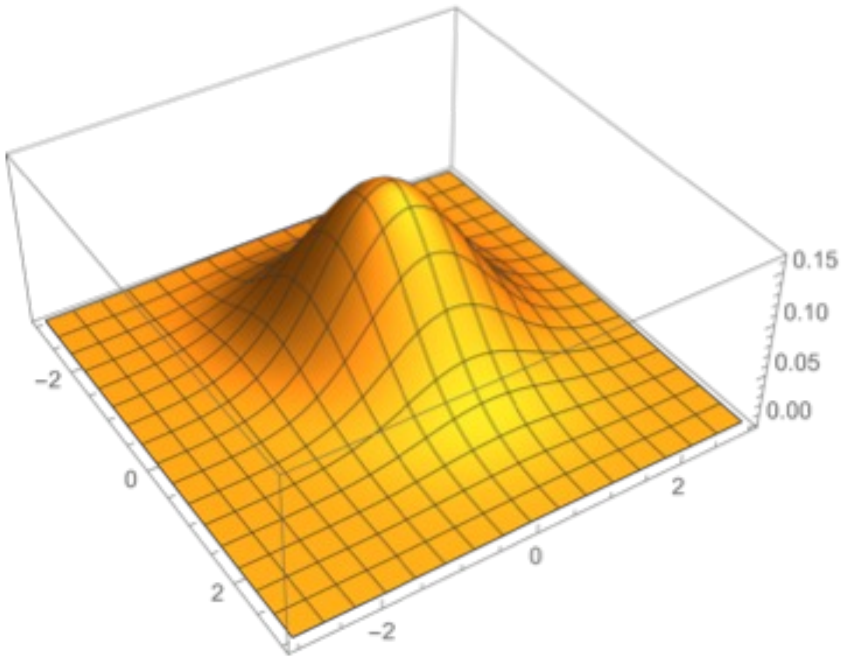
2-particle 1d harmonic, DNN



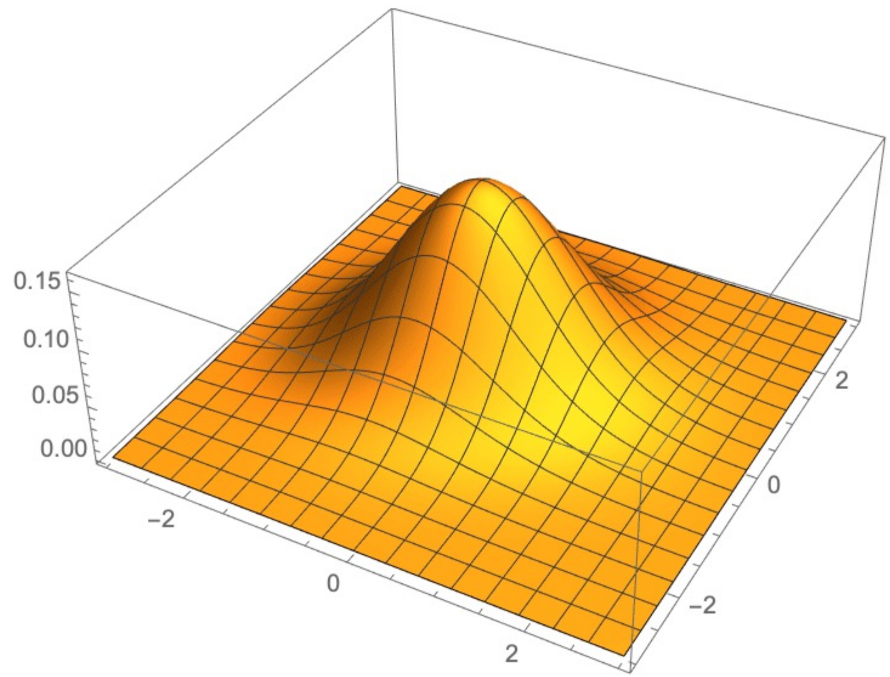
DNN with ReLU replacement

④. Discrete geometry

Region-wise linear approximation



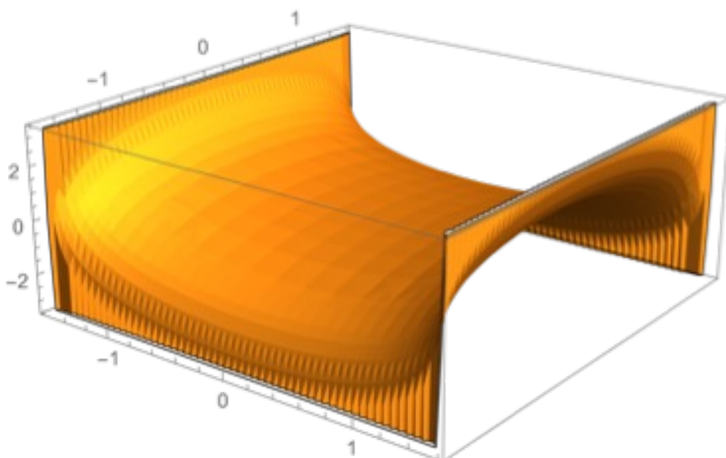
2-particle 1d harmonic, DNN



DNN with ReLU replacement

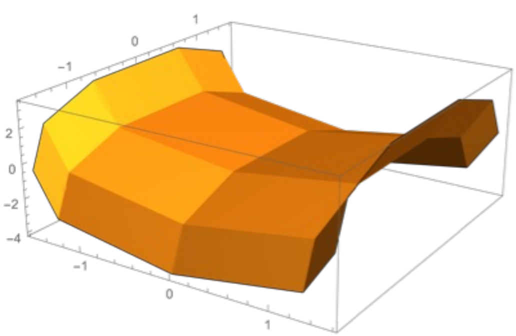
④. Discrete geometry

Working as a discrete geometry method

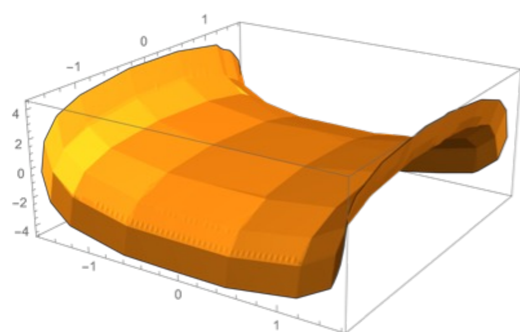


Scherk surface

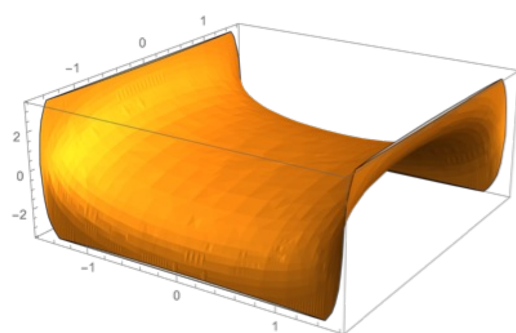
$$\log\left[\frac{\cos y}{\cos x}\right]$$



DNN 2-4-4-4-1



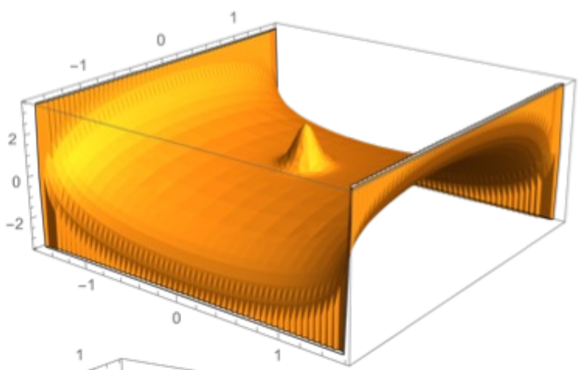
DNN 2-8-8-8-1

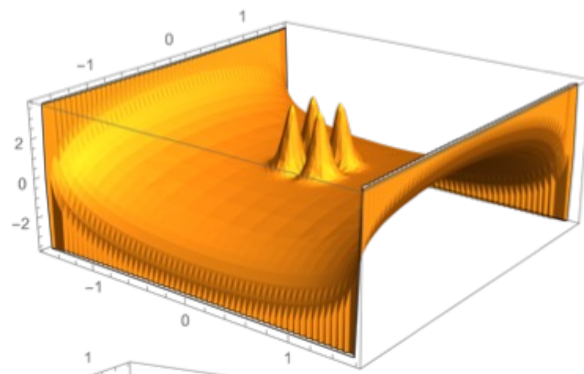


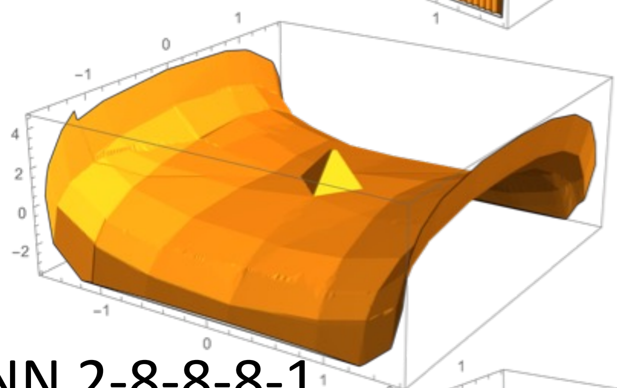
DNN 2-20-20-20-1

④ Discrete geometry

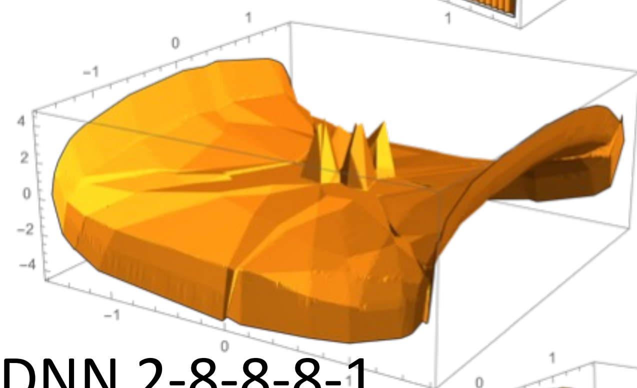
High curvature region automatically designed



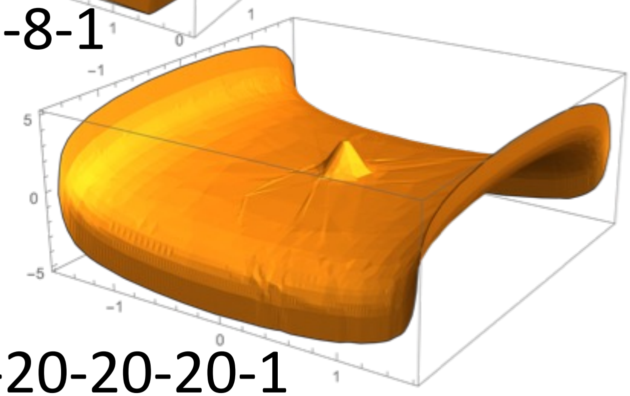




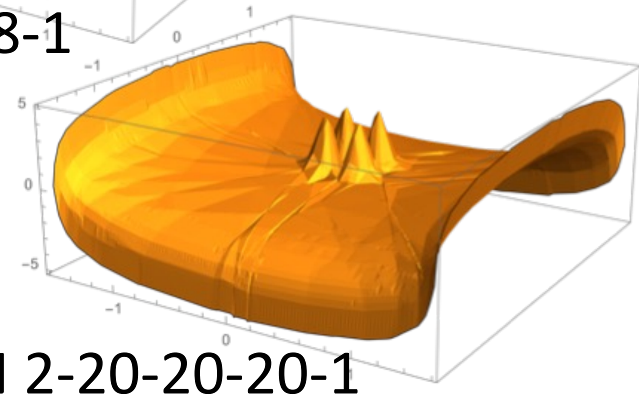
DNN 2-8-8-8-1



DNN 2-8-8-8-1



DNN 2-20-20-20-1



DNN 2-20-20-20-1

④. Discrete geometry

DNN giving optimal lattice, then?

Benefit

1. Computation-easy way to get discrete geometry
2. Works even when no smooth surface is given
3. Interpolates the smooth and the discrete

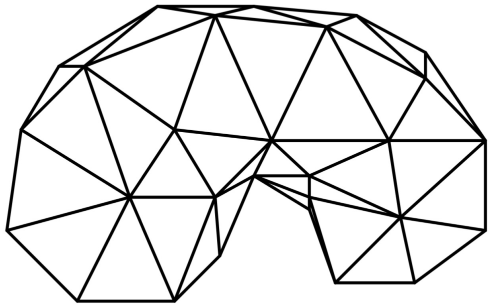
Quantum gravity!

1. Any space can be embedded as a surface in higher dim
2. Loss function can be Einstein equation
3. Weight quantization → Quantization of space
4. No spin network, thus different from AdS/DL

ML journey from QM to polytopes

1. Intro: NN quantum states — 4 pages
2. Solving QM with DNN — 5 pages
   ArXiv:2302.08965 [physics.comp-ph]
3. Multi-particle / interaction — 9 pages
   ArXiv:2302.08965 [physics.comp-ph]
4. Discrete geometry — 5 pages
   Unpublished (on-going work)
5. Neural polytopes — 6 pages
   ArXiv:2307.00721 [cs.LG]

Quantum gravity and discrete surfaces :
How can we sum all possible spacetimes?



Regge calculus

[Regge `61]

Fixed lattice architecture, variable lengths

Dynamical triangulation

[Ambjorn, Loll `98]

Randomly generated lattice architecture, fixed lengths

Approximating a sphere by neural network

Supervised data set :

points on a sphere,

$\mathcal{D} \equiv \{\vec{x}^{(i)} \to 1 \mid \vec{x}^{(i)} \in S^{d-1}\}$

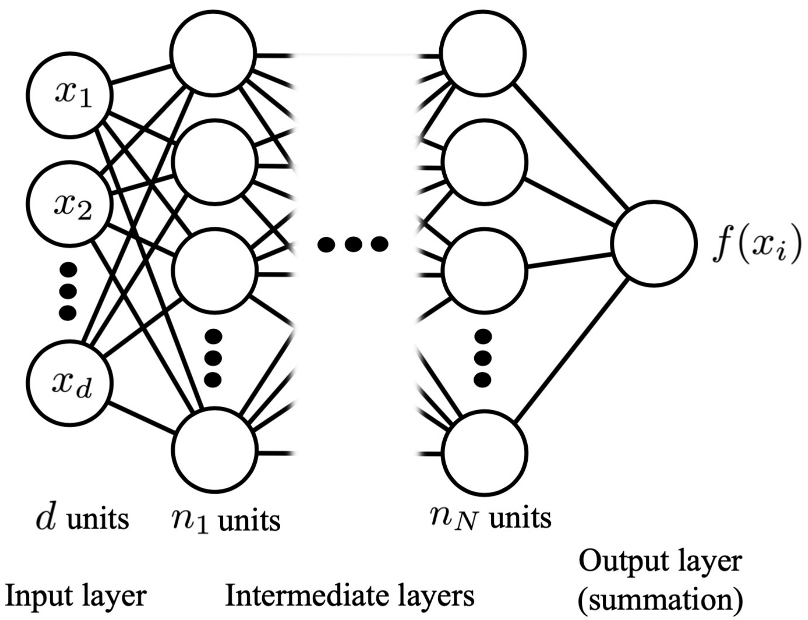
Architecture :

feedforward NN,

fully connected, no bias

Activation function :

Generalized ReLU $\varphi(x) = |x|^p$



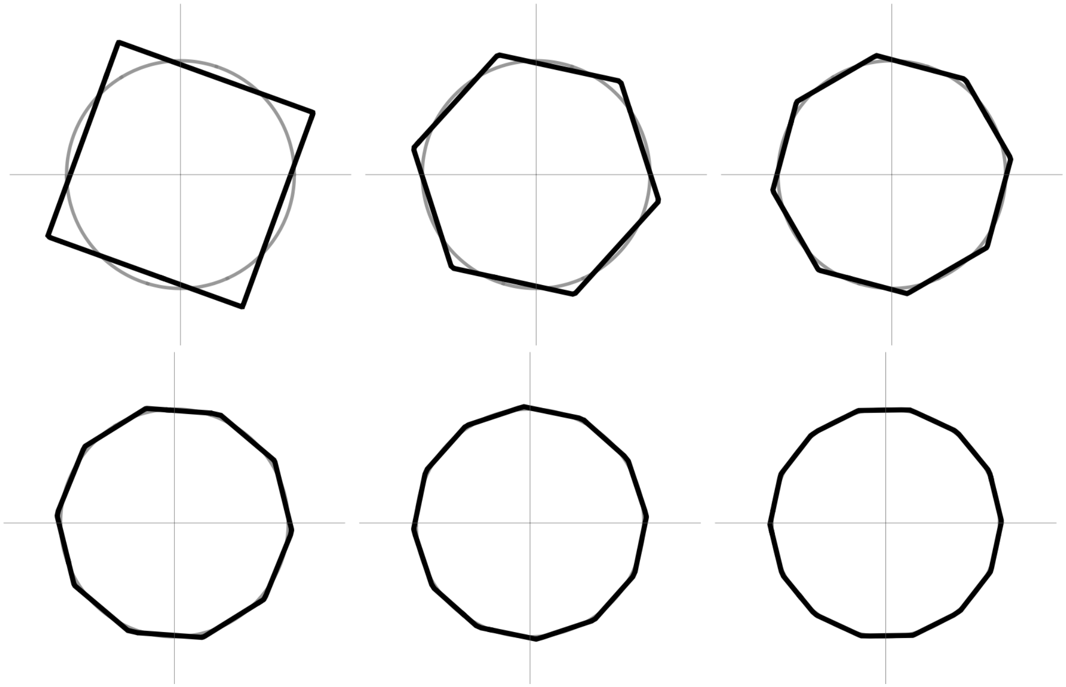
training

Neural d-polytopes of type $(n_1, \ldots, n_N; p_1, \ldots, p_N)$

||

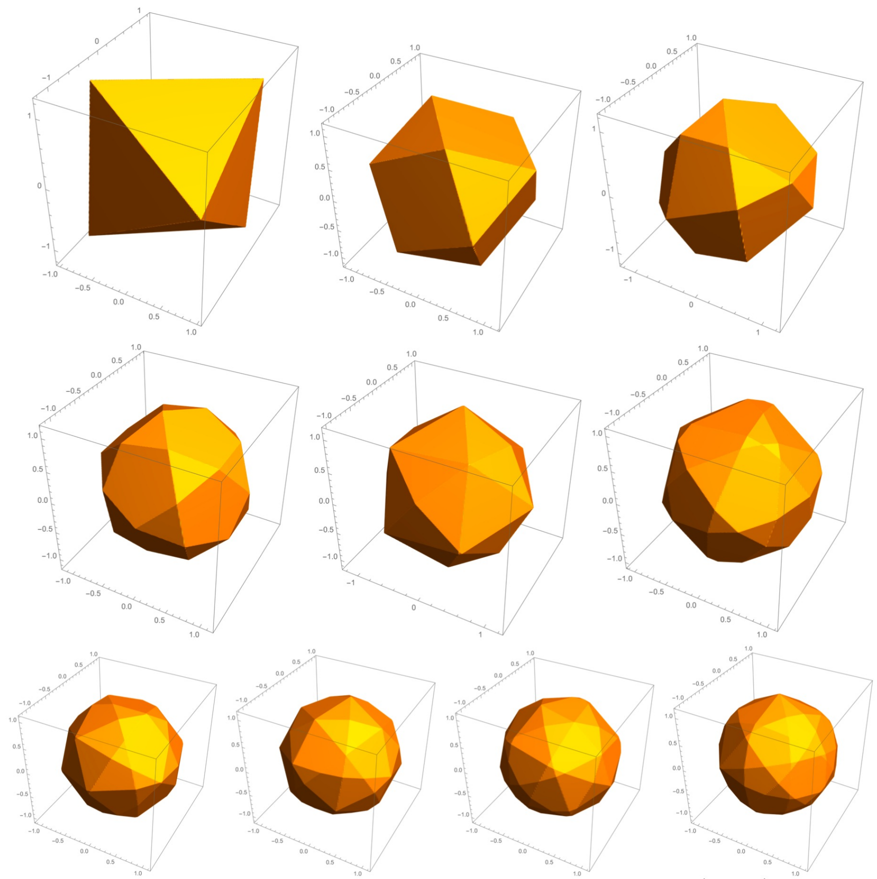
Cross section defined by $f(x_i) = 1$

[Result 1] Generative polytopes : successful discrete geometry by machine learning



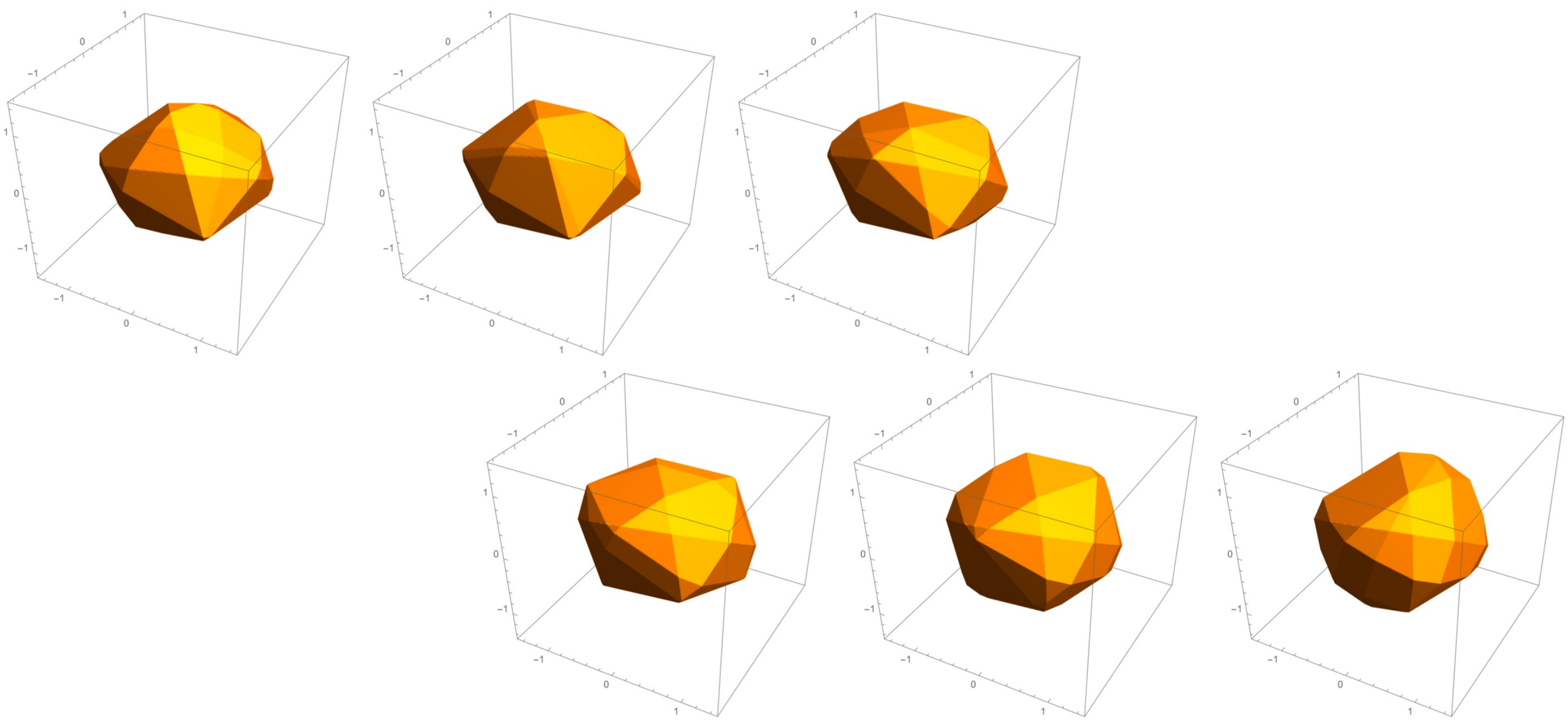
Neural 2-polytopes of type (n ; 1) for n = 2, 3, 4, 5, 6, 7.

[Result 1] Generative polytopes :
successful discrete geometry by machine learning



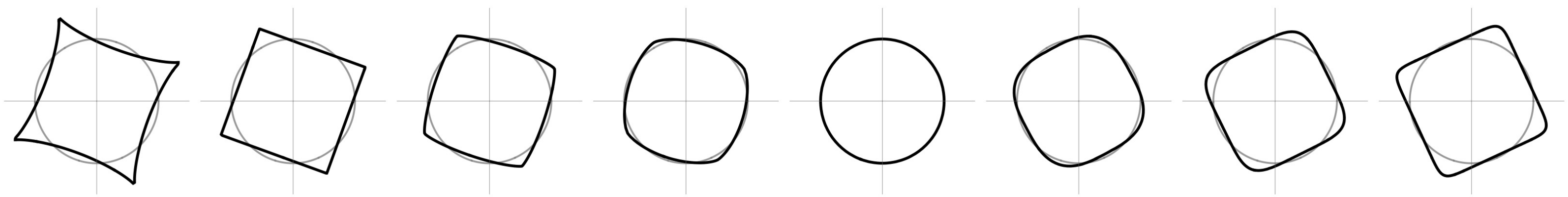
Neural 3-polytopes of type (n ; 1) for n = 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.

[Result 1] Generative polytopes :
successful discrete geometry by machine learning

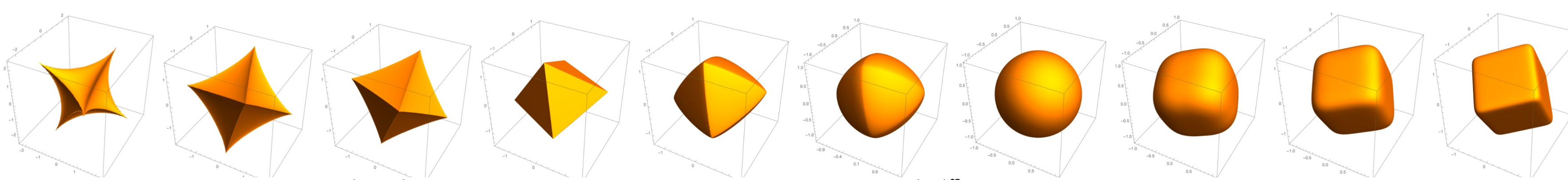


Neural 5-polytopes of type (8 ; 1) sliced at codimension-2 plane rotated gradually.

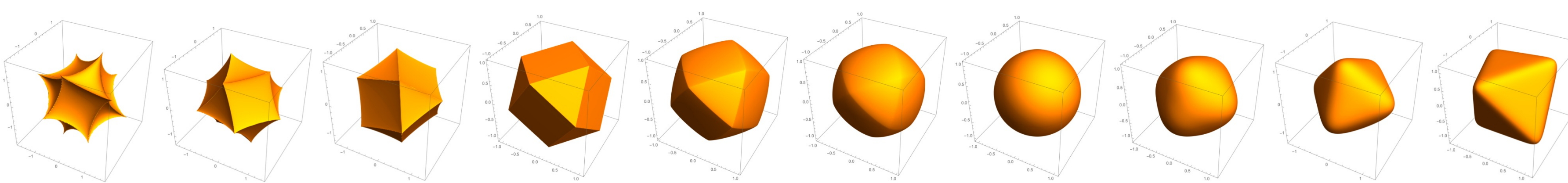
[Result 2] Neural polytopes :
sphere approximation with various activation functions



Neural polygons of type $(2;p)$, with the activation function is chosen as $|x|^p$, where $p = 0.8,\ 1.0,\ 1.2,\ 1.5,\ 2,\ 3,\ 5,\ 10$



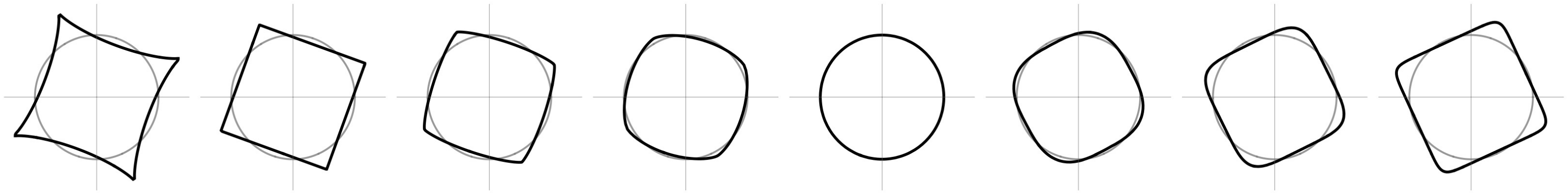
Neural polyhedra of type $(3;p)$, with the activation function is chosen as $|x|^p$, where $p = 0.6,\ 0.8,\ 0.9,\ 1.0,\ 1.2,\ 1.5,\ 2,\ 3,\ 5,\ 10$



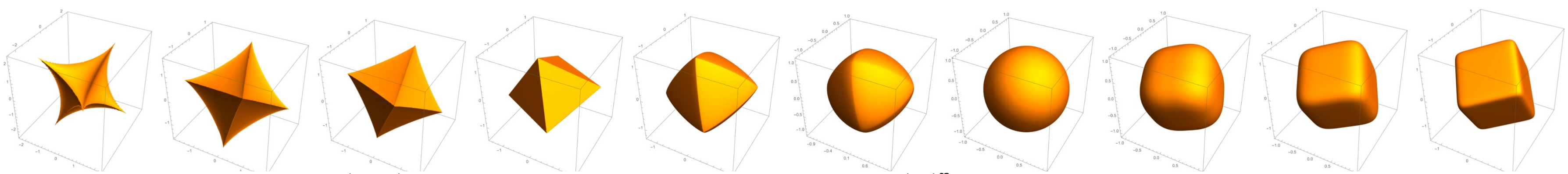
Neural polyhedra of type $(4;p)$, with the activation function is chosen as $|x|^p$, where $p = 0.6,\ 0.8,\ 0.9,\ 1.0,\ 1.2,\ 1.5,\ 2,\ 3,\ 5,\ 10$

[Result 2] Neural polytopes :

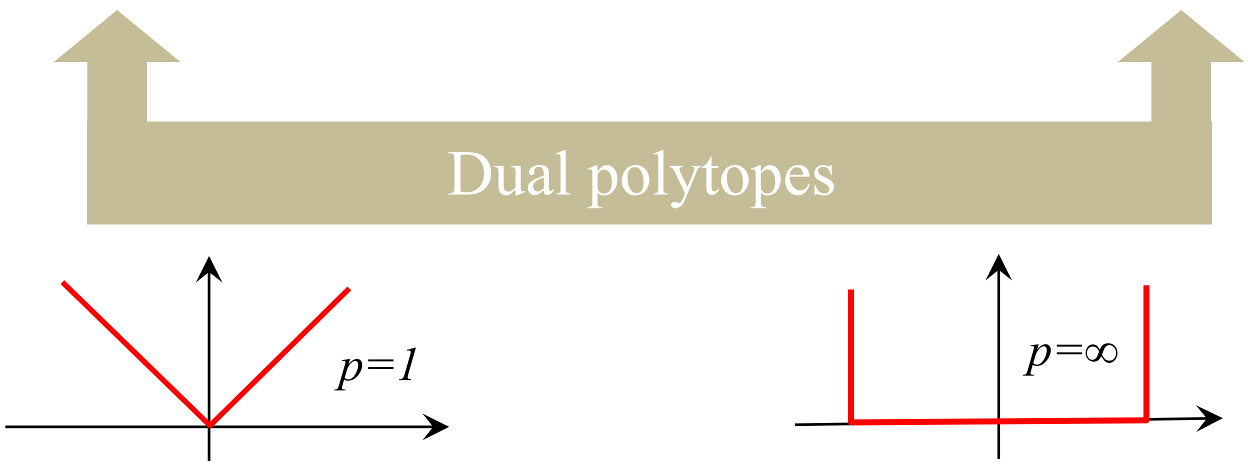
sphere approximation with various activation functions



Neural polygons of type $(2;p)$, with the activation function is chosen as $|x|^p$, where $p = 0.8, 1.0, 1.2, 1.5, 2, 3, 5, 10$



Neural polyhedra of type $(3;p)$, with the activation function is chosen as $|x|^p$, where $p = 0.6, 0.8, 0.9, 1.0, 1.2, 1.5, 2, 3, 5, 10$



ML journey from QM to polytopes

1. Intro: NN quantum states — 4 pages
2. Solving QM with DNN — 5 pages
   ArXiv:2302.08965 [physics.comp-ph]
3. Multi-particle / interaction — 9 pages
   ArXiv:2302.08965 [physics.comp-ph]
4. Discrete geometry — 5 pages
   Unpublished (on-going work)
5. Neural polytopes — 6 pages
   ArXiv:2307.00721 [cs.LG]