Simulating atomic nuclei in a quantum computer

arXiv:2302.03641

Axel Pérez-Obiol, Antonio Márquez, Javier Menéndez,
Arnau Rios, Artur Garcia, Bruno Juliá-Díaz

QUANTIC group

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Dept. Física quàntica i astrofísica

Universitat de Barcelona

Santiago, May 28th 2023

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA PRIMERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
Financiado por la Unión Europea
NextGenerationEU
Quantum SPAIN
Plan de Recuperación, Transformación y Resiliencia
RES
RED ESPAÑOLA DE SUPERCOMPUTACIÓN

Quantum simulation

Quantum many body problems are hard (exponential scaling)
(dammit! -R.F.)

molecules, crystals, nuclei, optical lattices, etc. ⟺ $\mathscr{H} = \sum_{ij} t_{ij} a_i^\dagger a_j + \sum_{ijkl} t_{ijkl} a_i^\dagger a_j^\dagger a_k a_l$

Quantum simulation

Quantum many body problems are hard (exponential scaling)
(dammit! -R.F.)

molecules, crystals, nuclei, optical lattices, etc. ⟺ $$\mathscr{H} = \sum_{ij} t_{ij} a_i^\dagger a_j + \sum_{ijkl} t_{ijkl} a_i^\dagger a_j^\dagger a_k a_l$$

Classical hardware

- Exact diagonalization (small systems)
- Integrable cases (very particular)
- Mean field, MC, PT… (approximations)

Quantum simulation

Quantum many body problems are hard (exponential scaling)
(dammit! -R.F.)

molecules, crystals, nuclei, optical lattices, etc.

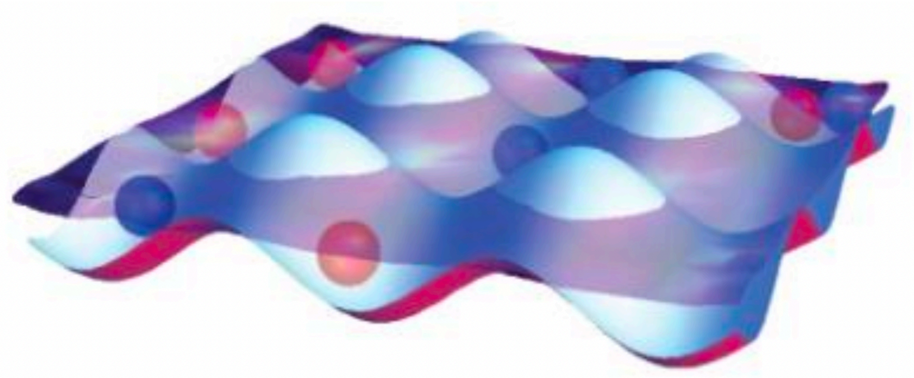
$$\mathscr{H} = \sum_{ij} t_{ij}\, a_i^\dagger a_j + \sum_{ijkl} t_{ijkl}\, a_i^\dagger a_j^\dagger a_k a_l$$

Classical hardware

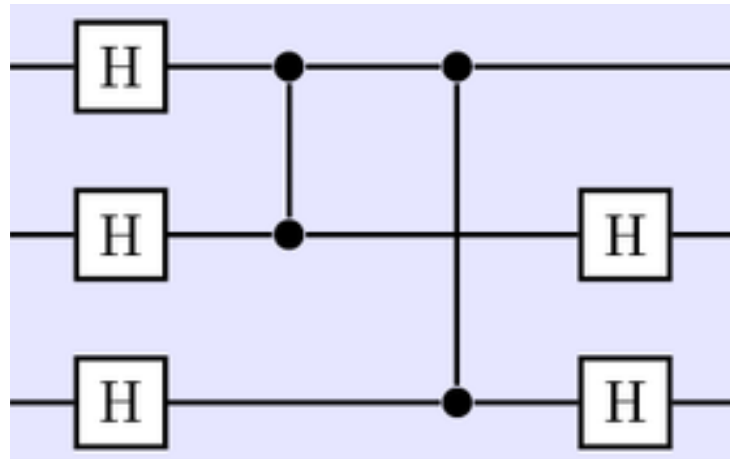
- Exact diagonalization (small systems)
- Integrable cases (very particular)
- Mean field, MC, PT… (approximations)

Quantum hardware

- quantum simulators



- digital quantum computers



Nuclear shell-model

1. Primordial shell model (IPM, naive):
   - ➡ Mean field:

     $$V(r) = \frac{1}{2}\hbar\omega r^2 + D\,\vec{l}^{\,2} + C\,\vec{l}\cdot\vec{s}$$

   - ➡ Predicts magic numbers
   - ➡ Valence space + effective interactions

Nuclear shell-model

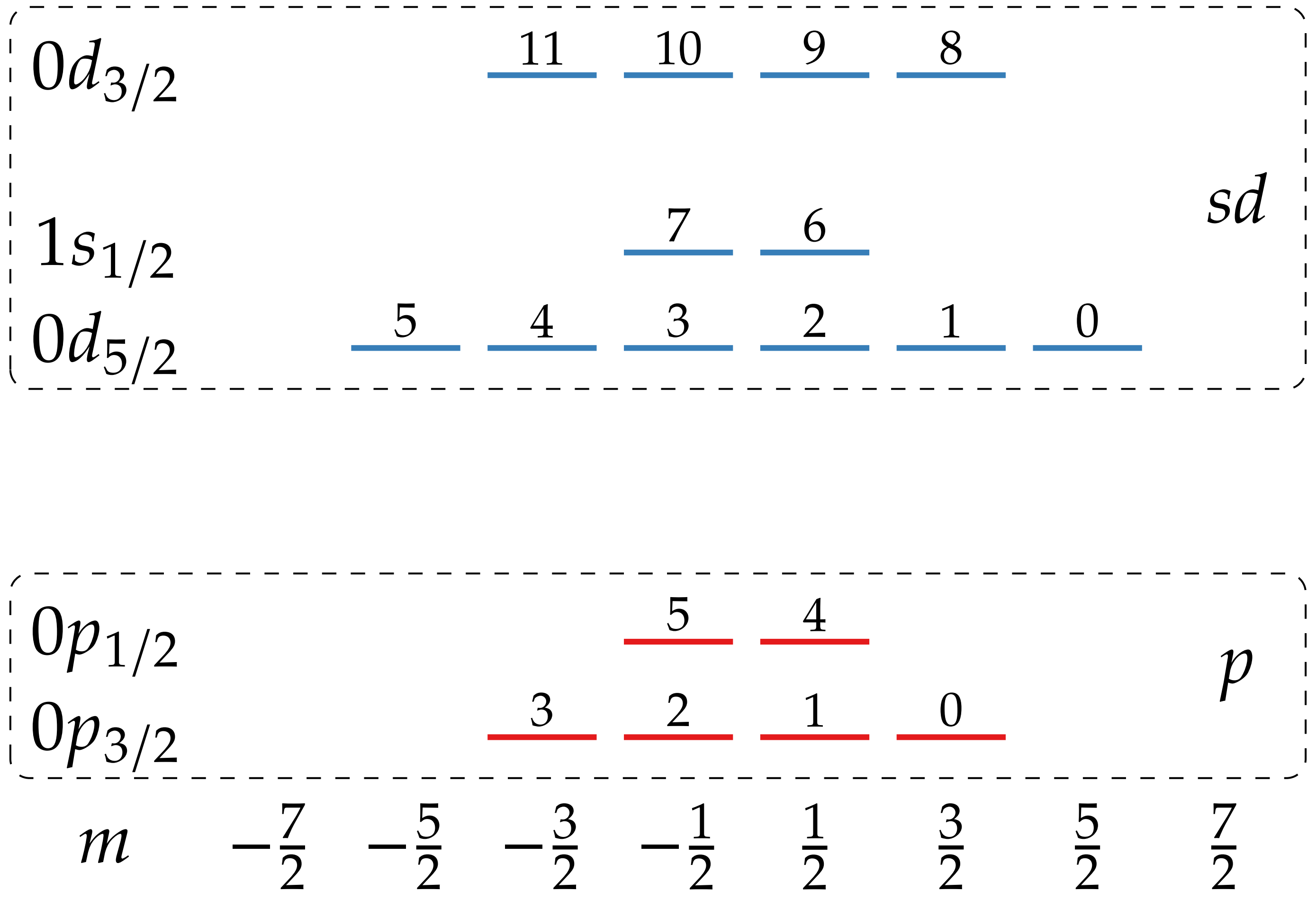
1. Primordial shell model (IPM, naive):

   ➡ Mean field:

   $$V(r) = \frac{1}{2}\hbar\omega r^2 + D\,\vec{l}^2 + C\,\vec{l}\cdot\vec{s}$$

   ➡ Predicts magic numbers

   ➡ Valence space + effective interactions



Nuclear shell-model

1. Primordial shell model (IPM, naive):
   - ➡ Mean field:

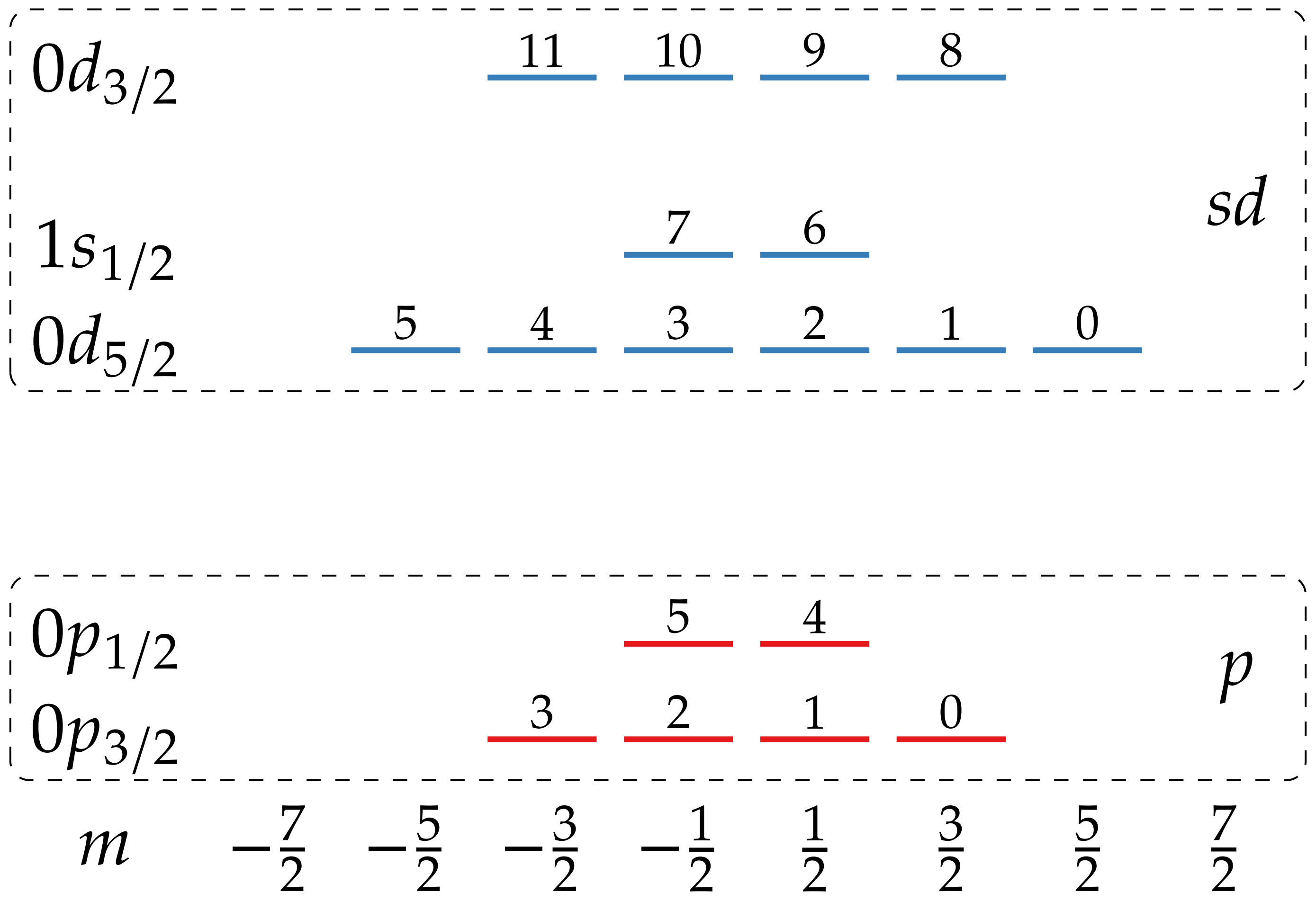
     $$V(r) = \frac{1}{2}\hbar\omega r^2 + D\,\vec{l}^2 + C\,\vec{l}\cdot\vec{s}$$

   - ➡ Predicts magic numbers
   - ➡ Valence space + effective interactions

2. Interaction shell model:
   - ➡ Mean field + residual two-body interactions:

     $$\mathscr{H} = \sum_{ij} K_{ij}\, a_i^\dagger a_j + \sum_{ijkl} V_{ijkl}\, a_i^\dagger a_j^\dagger a_k a_l$$

   - ➡ Diagonalization problem



Jordan-Wigner mapping:

$$a_j^\dagger = \prod_{k=0}^{j-1} Z_k \frac{1}{2}(X_j - iY_j)$$

Nuclear shell-model

1. Primordial shell model (IPM, naive):

   ➡ Mean field:

   $$V(r) = \frac{1}{2}\hbar\omega r^2 + D\,\vec{l}^{\,2} + C\,\vec{l}\cdot\vec{s}$$

   ➡ Predicts magic numbers

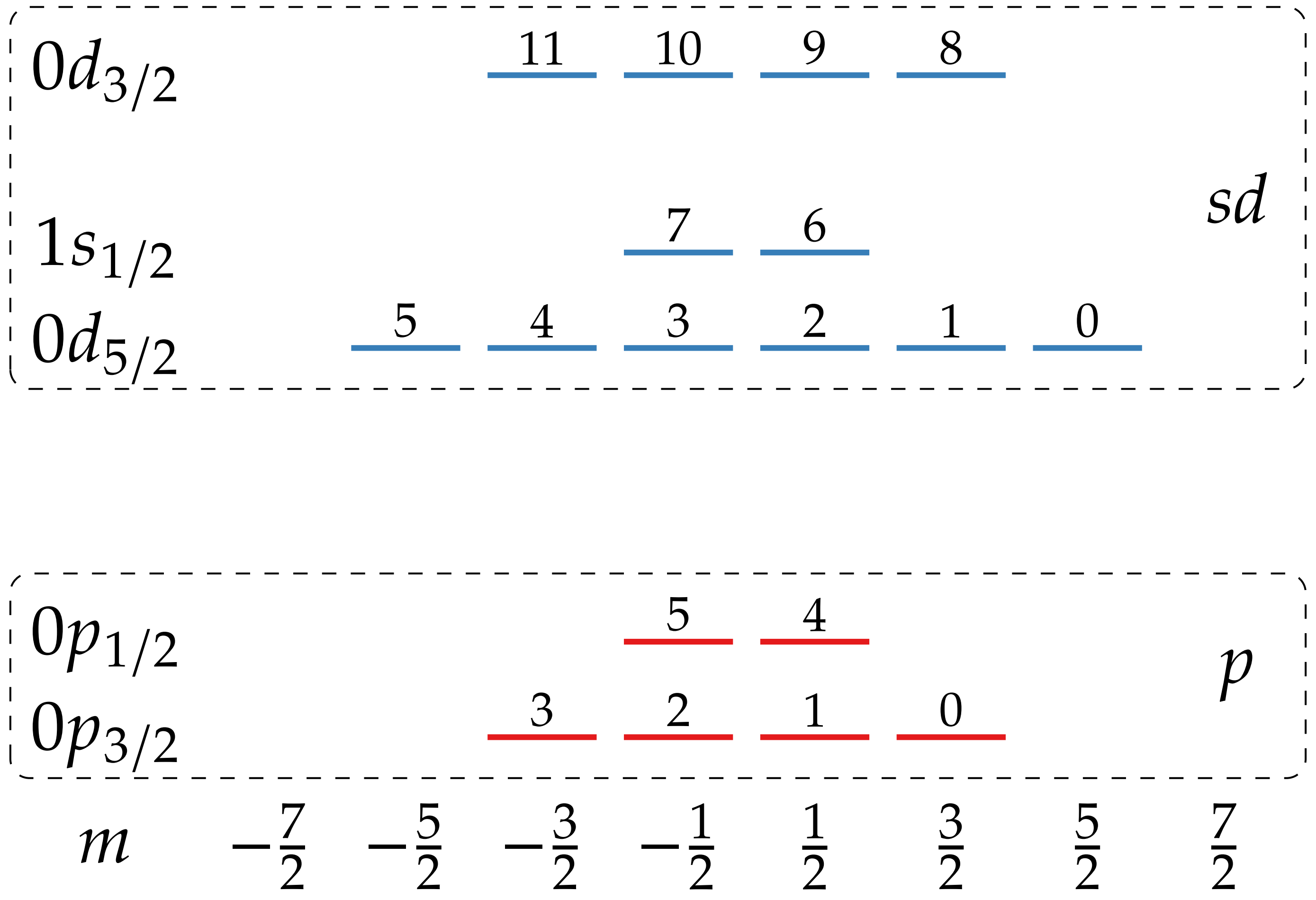
   ➡ Valence space + effective interactions

2. Interaction shell model:

   ➡ Mean field + residual two-body interactions:

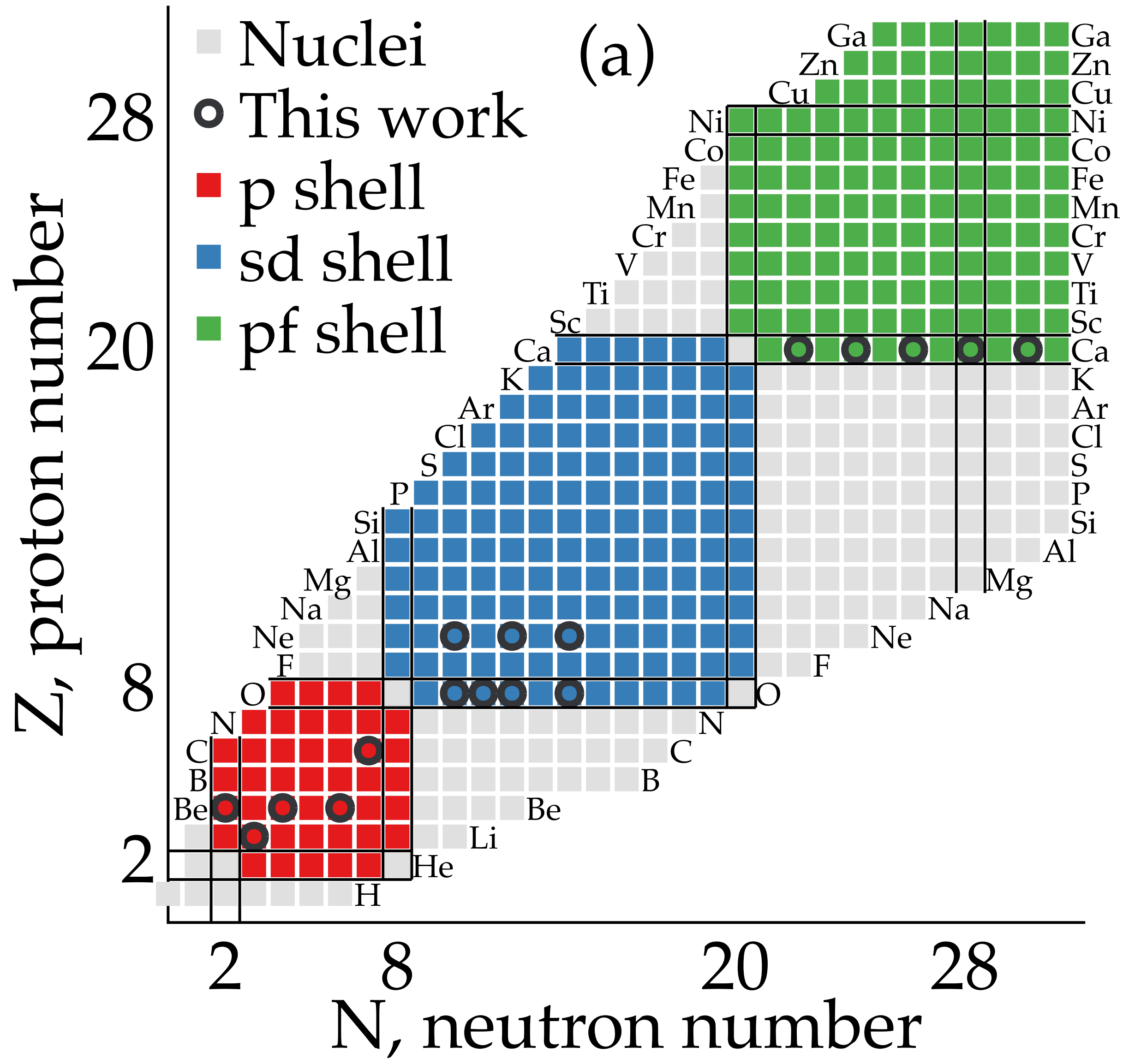
   $$\mathscr{H} = \sum_{ij} K_{ij}\, a_i^\dagger a_j + \sum_{ijkl} V_{ijkl}\, a_i^\dagger a_j^\dagger a_k a_l$$

   ➡ Diagonalization problem



Jordan-Wigner mapping:

$$a_j^\dagger = \prod_{k=0}^{j-1} Z_k \frac{1}{2}(X_j - iY_j)$$



ADAPT-VQE

1. (Regular) variational quantum eigensolver :

   ➡ Parametrized, trial wave function (hardware efficient / physics inspired)

   ➡ Optimize cost function: $E_{\text{ADAPT}} = \min_{\theta_k} \frac{\langle \Psi(\theta_k) | H | \Psi(\theta_k) \rangle}{\langle \Psi(\theta_k) | \Psi(\theta_k) \rangle}$

ADAPT-VQE

1. (Regular) variational quantum eigensolver :

   ➡ Parametrized, trial wave function (hardware efficient / physics inspired)

   ➡ Optimize cost function: $E_{\text{ADAPT}} = \min_{\theta_k} \frac{\langle \Psi(\theta_k) | H | \Psi(\theta_k) \rangle}{\langle \Psi(\theta_k) | \Psi(\theta_k) \rangle}$

2. Adaptive VQE:

   1. Choose a pool of operators: $A_k = i(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)$

Grimsley et al., *Nat. comm.* **10**, 1–9 (2019)

ADAPT-VQE

1. (Regular) variational quantum eigensolver :

   ➡ Parametrized, trial wave function (hardware efficient / physics inspired)

   ➡ Optimize cost function: $E_{\text{ADAPT}} = \min_{\theta_k} \frac{\langle \Psi(\theta_k) | H | \Psi(\theta_k) \rangle}{\langle \Psi(\theta_k) | \Psi(\theta_k) \rangle}$

2. Adaptive VQE:

   1. Choose a pool of operators: $A_k = i(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)$

   2. Start from reference state: $|\psi_0\rangle = a_{i_1}^\dagger \ldots a_{i_{N_p}}^\dagger |0\rangle$

Grimsley et al., *Nat. comm.* **10**, 1–9 (2019)

ADAPT-VQE

1. (Regular) variational quantum eigensolver :

   ➡ Parametrized, trial wave function (hardware efficient / physics inspired)

   ➡ Optimize cost function: $E_{\text{ADAPT}} = \min_{\theta_k} \frac{\langle \Psi(\theta_k) | H | \Psi(\theta_k) \rangle}{\langle \Psi(\theta_k) | \Psi(\theta_k) \rangle}$

2. Adaptive VQE:

   1. Choose a pool of operators: $A_k = i(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)$

   2. Start from reference state: $|\psi_0\rangle = a_{i_1}^\dagger \ldots a_{i_{N_p}}^\dagger |0\rangle$

   3. Add operator from the pool with the largest gradient:
      $|\psi(\theta_1)\rangle_{\text{ADAPT}} = e^{i\theta_1 A_1} |\psi_0\rangle$, with maximum $\left. \frac{\partial E^{(n)}}{\partial \theta_k} \right|_{\theta_k = 0}$

Grimsley et al., *Nat. comm.* **10**, 1–9 (2019)

ADAPT-VQE

1. (Regular) variational quantum eigensolver :
   - ➡ Parametrized, trial wave function (hardware efficient / physics inspired)
   - ➡ Optimize cost function: $E_{\text{ADAPT}} = \min_{\theta_k} \frac{\langle \Psi(\theta_k) | H | \Psi(\theta_k) \rangle}{\langle \Psi(\theta_k) | \Psi(\theta_k) \rangle}$
2. Adaptive VQE:
   1. Choose a pool of operators: $A_k = i(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)$
   2. Start from reference state: $|\psi_0\rangle = a_{i_1}^\dagger \ldots a_{i_{N_p}}^\dagger |0\rangle$
   3. Add operator from the pool with the largest gradient:
      $|\psi(\theta_1)\rangle_{\text{ADAPT}} = e^{i\theta_1 A_1} |\psi_0\rangle$, with maximum $\left.\frac{\partial E^{(n)}}{\partial \theta_k}\right|_{\theta_k = 0}$
   4. Optimize all parameters: $|\psi(\theta_k)\rangle_{\text{ADAPT}} = \prod_{k=1}^{n} e^{i\theta_k A_k} |\psi_0\rangle$

Grimsley et al., *Nat. comm.* **10**, 1–9 (2019)

ADAPT-VQE

1. (Regular) variational quantum eigensolver :

   ➡ Parametrized, trial wave function (hardware efficient / physics inspired)

   ➡ Optimize cost function: $E_{\text{ADAPT}} = \min_{\theta_k} \frac{\langle \Psi(\theta_k) | H | \Psi(\theta_k) \rangle}{\langle \Psi(\theta_k) | \Psi(\theta_k) \rangle}$

2. Adaptive VQE:

   1. Choose a pool of operators: $A_k = i(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)$

   2. Start from reference state: $|\psi_0\rangle = a_{i_1}^\dagger \ldots a_{i_{N_p}}^\dagger |0\rangle$

   3. Add operator from the pool with the largest gradient:
      $|\psi(\theta_1)\rangle_{\text{ADAPT}} = e^{i\theta_1 A_1} |\psi_0\rangle$, with maximum $\left. \frac{\partial E^{(n)}}{\partial \theta_k} \right|_{\theta_k = 0}$

   4. Optimize all parameters: $|\psi(\theta_k)\rangle_{\text{ADAPT}} = \prod_{k=1}^{n} e^{i\theta_k A_k} |\psi_0\rangle$

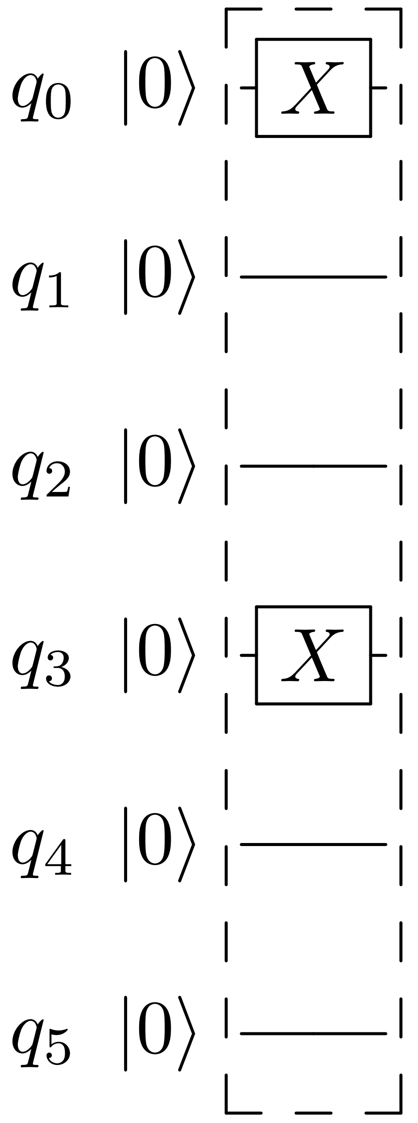
iterate

Grimsley et al., *Nat. comm.* **10**, 1–9 (2019)

Circuit

1. Initial state preparation, e.g., $|\psi_0\rangle = a_0^\dagger a_3^\dagger |0\rangle = X_0 X_3 |0\rangle$

$a_0^\dagger a_3^\dagger |0\rangle$



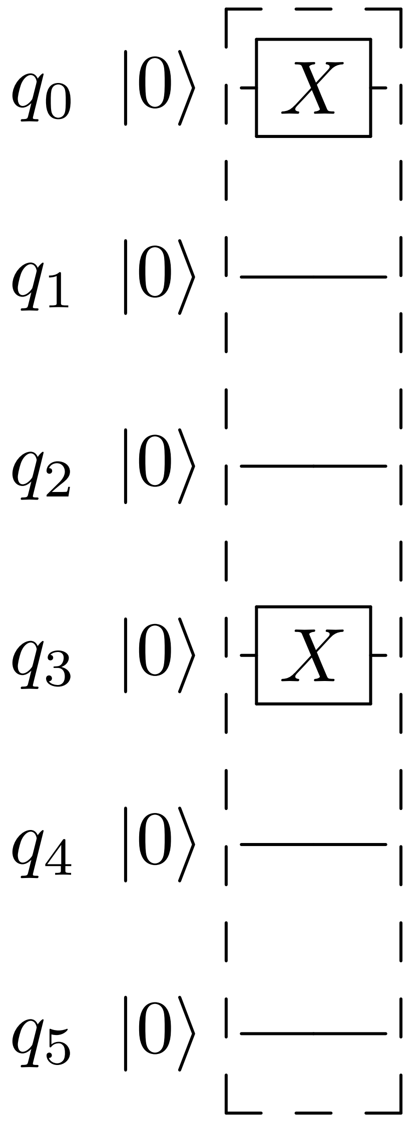
$O(1)$

Circuit

1. Initial state preparation, e.g.,
$|\psi_0\rangle = a_0^\dagger a_3^\dagger |0\rangle = X_0 X_3 |0\rangle$

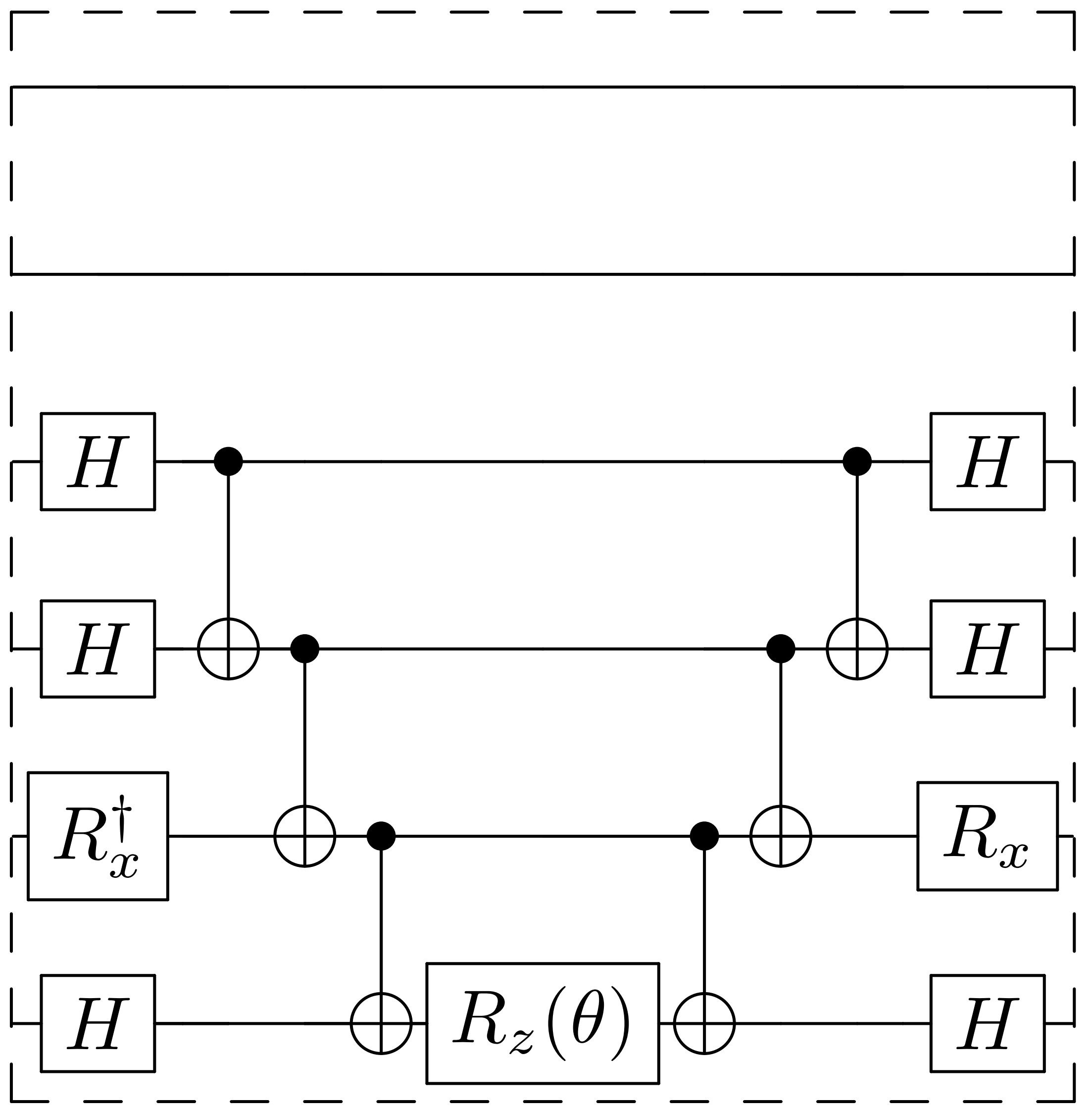
2. Each ansatz layer implemented with the staircase algorithm:
$e^{-\theta(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)} = e^{-i\frac{\theta}{2}X_2X_3Y_4X_5} \times (\ldots)$

$a_0^\dagger a_3^\dagger |0\rangle$



$O(1)$

$e^{-i\frac{\theta}{2}X_2X_3Y_4X_5}$



$\times 8 \times N_{layers}$

$O(N_q)$

Circuit

1. Initial state preparation, e.g.,

$|\psi_0\rangle = a_0^\dagger a_3^\dagger |0\rangle = X_0 X_3 |0\rangle$

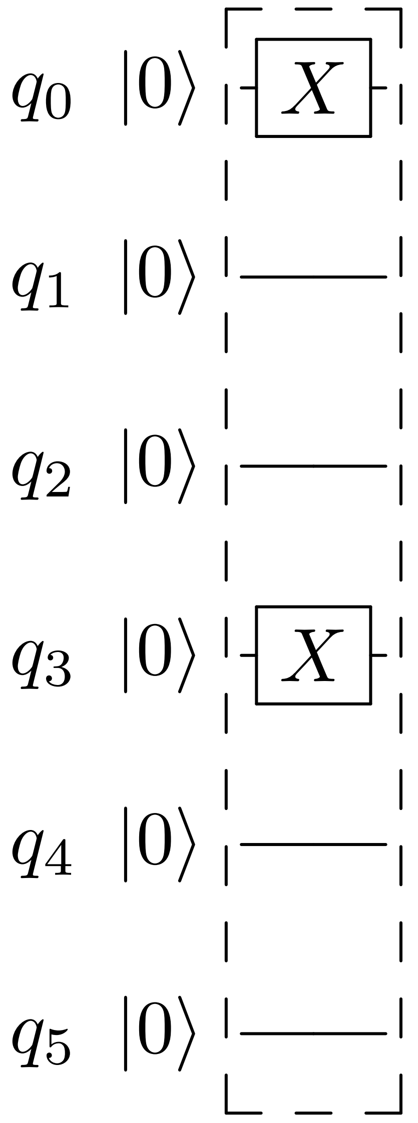
2. Each ansatz layer implemented with the staircase algorithm:

$e^{-\theta(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)} = e^{-i\frac{\theta}{2}X_2X_3Y_4X_5} \times (\dots)$

3. Measurement

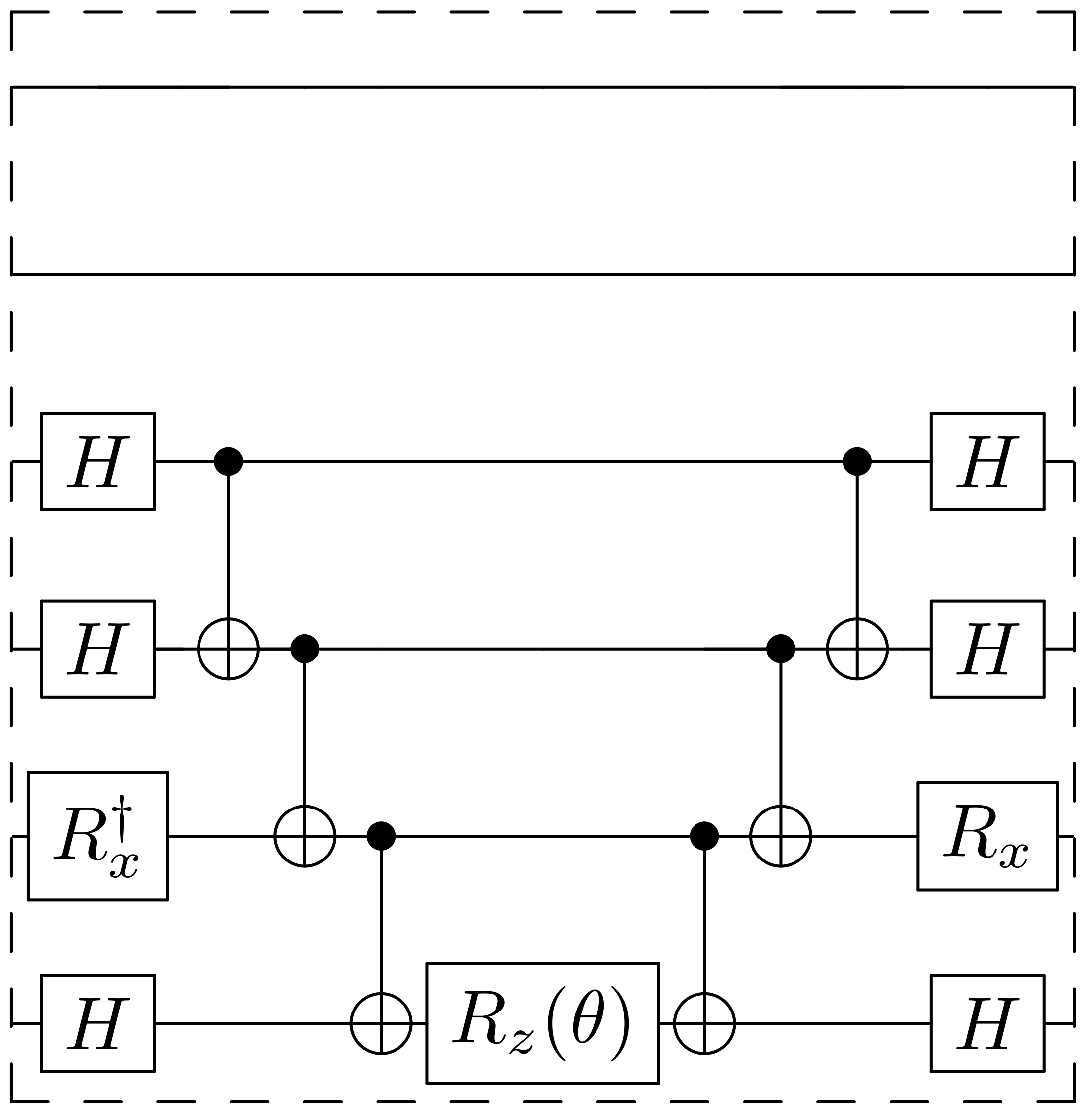
- ➡ $h = a_p^\dagger a_q^\dagger a_r a_s + a_r^\dagger a_s^\dagger a_p a_q$
$\propto |0011\rangle\langle 1100| + |1100\rangle\langle 0011|$
- ➡ $\langle\psi|h|\psi\rangle = \langle\psi|M^\dagger(MhM^\dagger)M|\psi\rangle$
$= p_{1100} - p_{0011}$

$a_0^\dagger a_3^\dagger |0\rangle$



$O(1)$

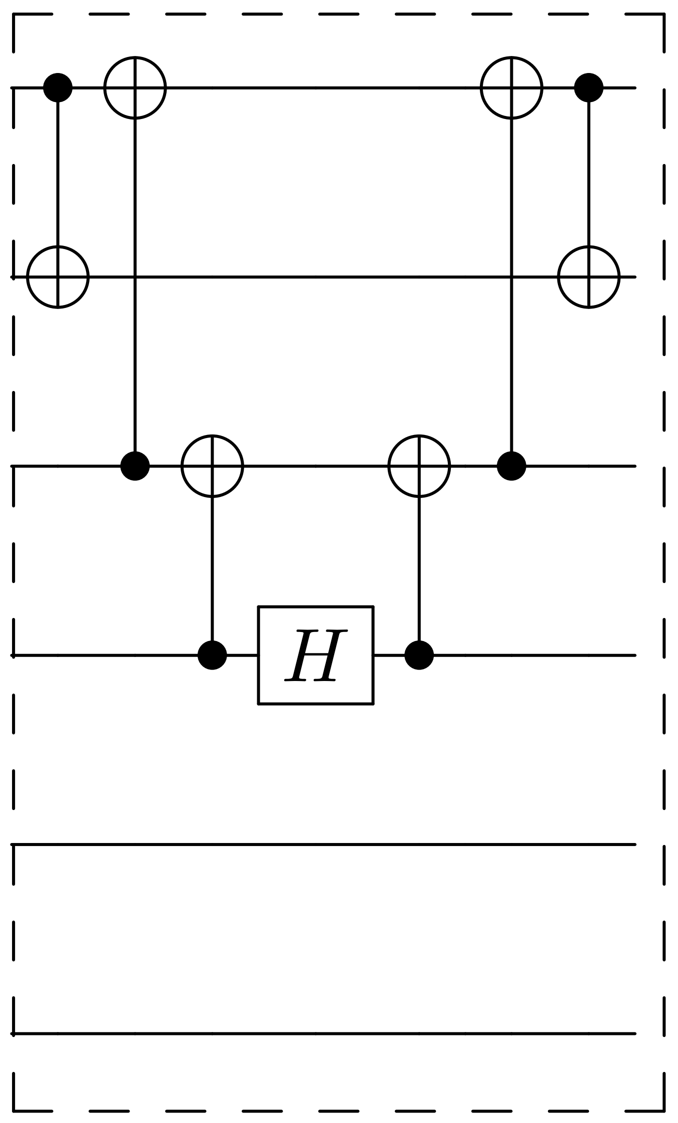
$e^{-i\frac{\theta}{2}X_2X_3Y_4X_5}$



$\times 8 \times N_{layers}$

$O(N_q)$

M



$O(1)$

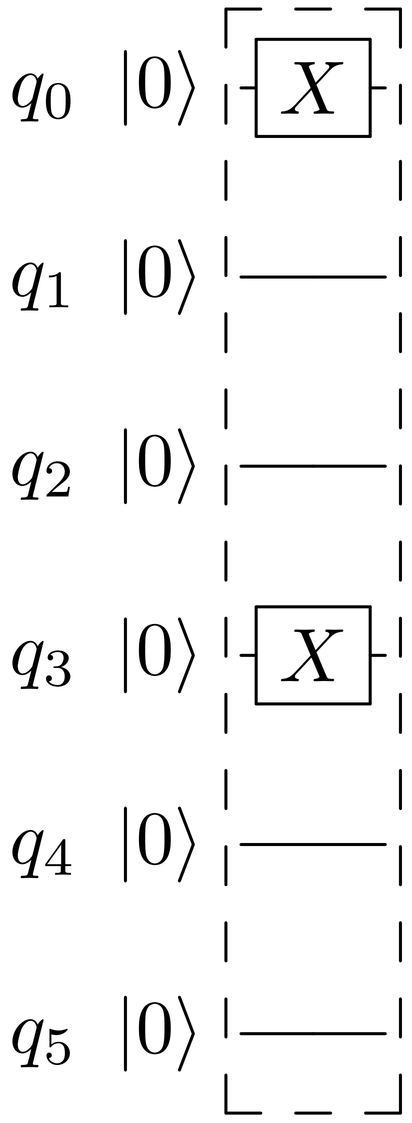
Circuit

1. Initial state preparation, e.g.,
$|\psi_0\rangle = a_0^\dagger a_3^\dagger |0\rangle = X_0 X_3 |0\rangle$

2. Each ansatz layer implemented with the staircase algorithm:
$e^{-\theta(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)} = e^{-i\frac{\theta}{2}X_2X_3Y_4X_5} \times (\ldots)$

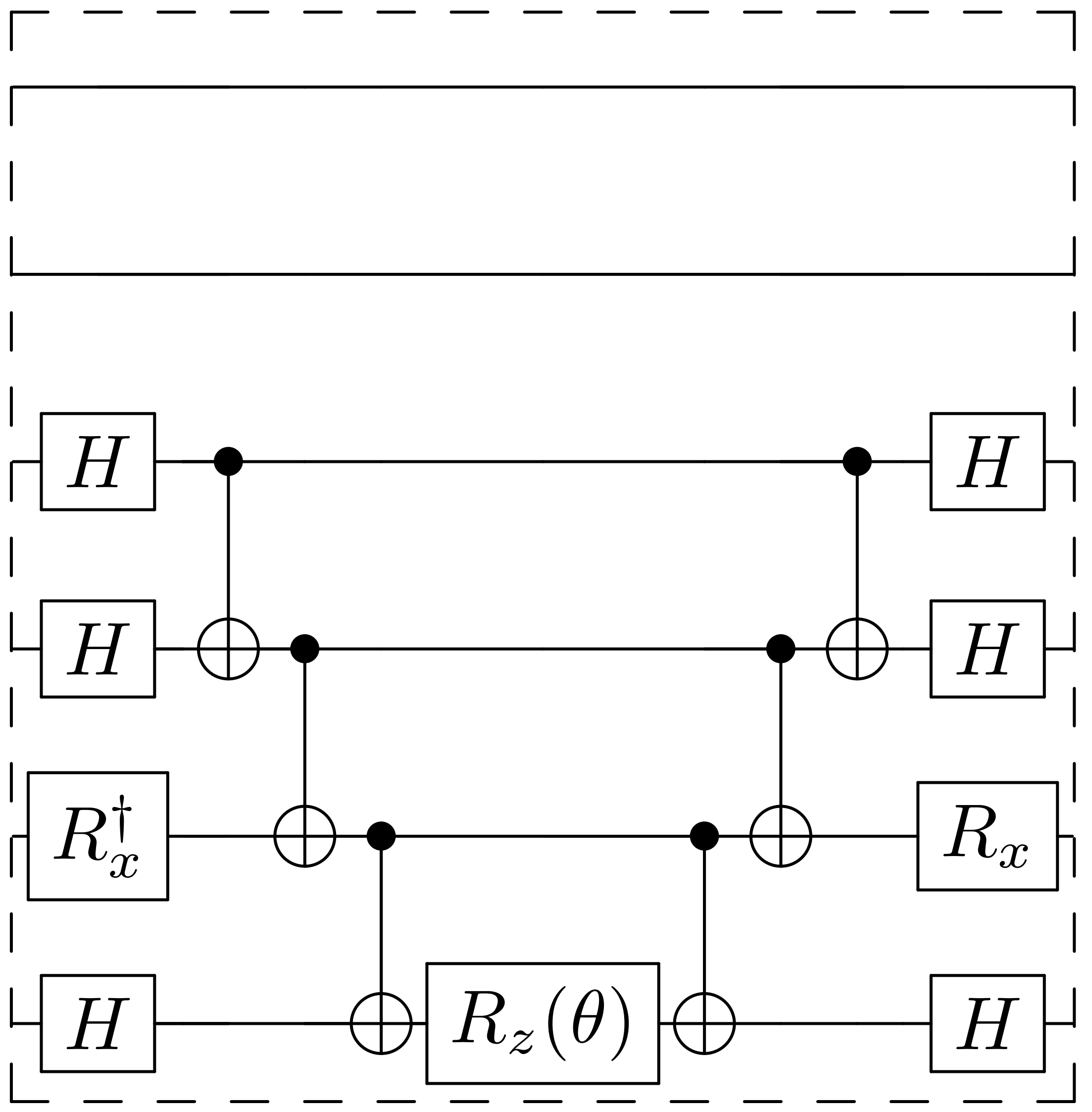
3. Measurement
➡ $h = a_p^\dagger a_q^\dagger a_r a_s + a_r^\dagger a_s^\dagger a_p a_q \propto |0011\rangle\langle 1100| + |1100\rangle\langle 0011|$
➡ $\langle\psi|h|\psi\rangle = \langle\psi|M^\dagger(MhM^\dagger)M|\psi\rangle = p_{1100} - p_{0011}$

$a_0^\dagger a_3^\dagger |0\rangle$



$O(1)$

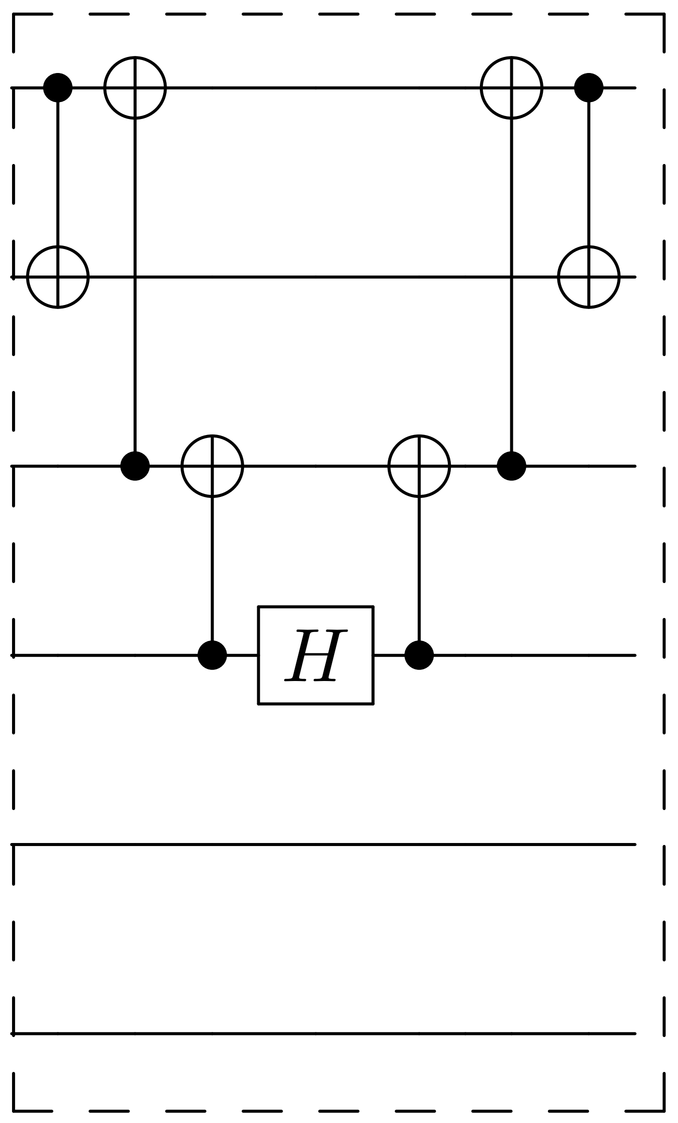
$e^{-i\frac{\theta}{2}X_2X_3Y_4X_5}$



$\times 8 \times N_{layers}$

$O(N_q)$

M



$O(1)$

How many layers?

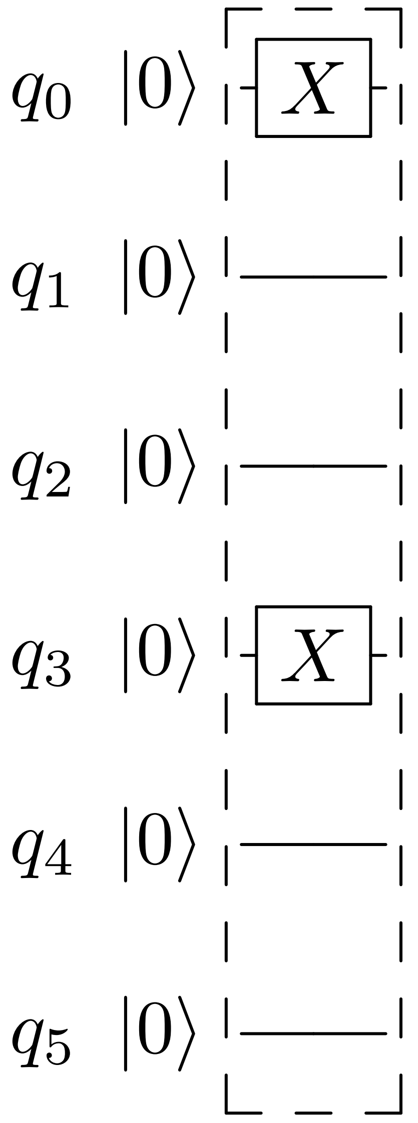
Circuit

1. Initial state preparation, e.g.,
$|\psi_0\rangle = a_0^\dagger a_3^\dagger |0\rangle = X_0 X_3 |0\rangle$

2. Each ansatz layer implemented with the staircase algorithm:
$e^{-\theta(a_p^\dagger a_q^\dagger a_r a_s - a_r^\dagger a_s^\dagger a_p a_q)} = e^{-i\frac{\theta}{2}X_2X_3Y_4X_5} \times (\ldots)$

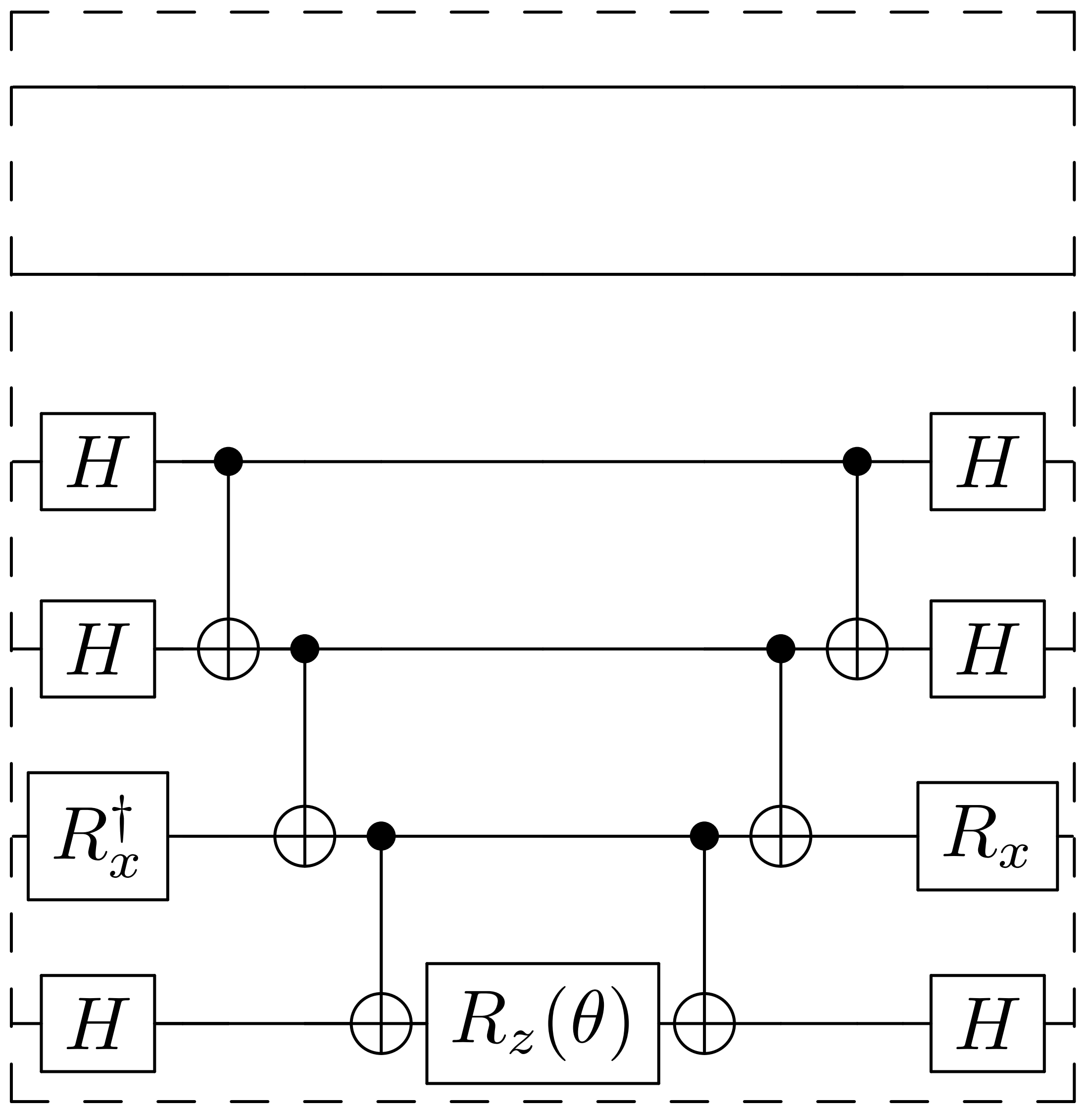
3. Measurement
- ➡ $h = a_p^\dagger a_q^\dagger a_r a_s + a_r^\dagger a_s^\dagger a_p a_q \propto |0011\rangle\langle 1100| + |1100\rangle\langle 0011|$
- ➡ $\langle\psi|h|\psi\rangle = \langle\psi|M^\dagger(MhM^\dagger)M|\psi\rangle = p_{1100} - p_{0011}$

$a_0^\dagger a_3^\dagger |0\rangle$



$O(1)$

$e^{-i\frac{\theta}{2}X_2X_3Y_4X_5}$

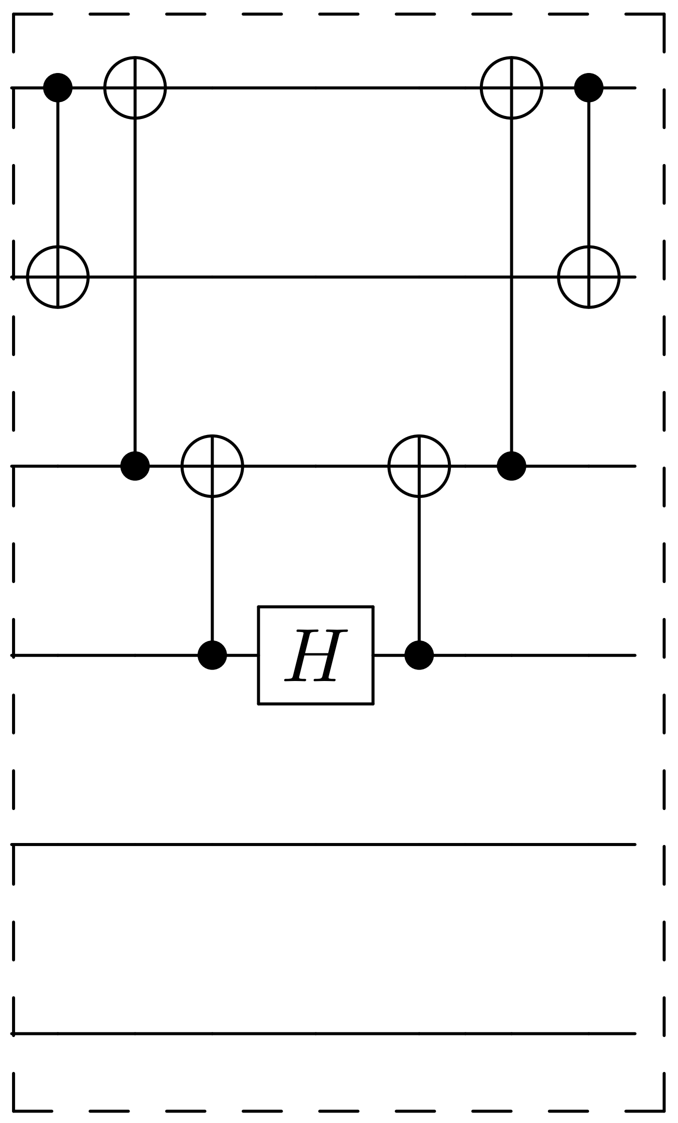


$\times 8 \times N_{layers}$

$O(N_q)$

How many layers?

M

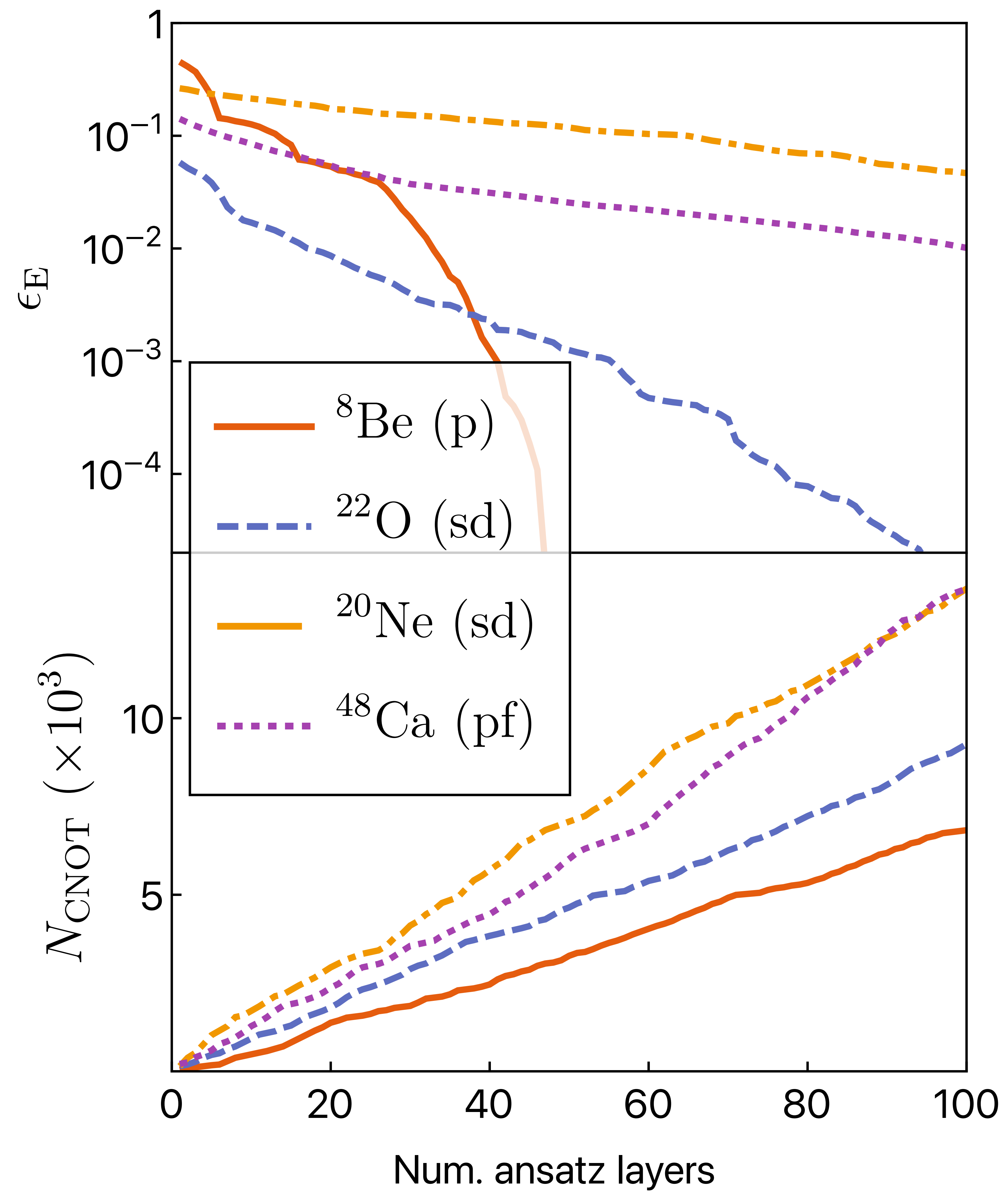


$O(1)$

How many measurements?

Number of parameters/CNOTs needed

quantum circuit simulation (qibo, GPUs, Marenostrum 4)



shell	N_{qb}	N_{SD}	nucleus	N_{layers}	ϵ_E bound	N_C (bound)
	6	5	^{6}Be	2	10^{-8}	42 (80)
	12	10	^{6}Li	9	10^{-7}	92 (176)
p		53	^{8}Be	48	10^{-7}	68 (176)
		51	^{10}Be	48	10^{-7}	62 (176)
		42	^{13}C	17	10^{-5}	77 (176)
	12	14	^{18}O	5	10^{-6}	99 (176)
		74	^{19}O	32	10^{-6}	85 (176)
		81	^{20}O	70	10^{-6}	98 (176)
sd		142	^{22}O	117	10^{-6}	93 (176)
	24	640	^{20}Ne	167	2×10^{-2}	137 (368)
		4206	^{22}Ne	236	2×10^{-2}	137 (368)
		7562	^{24}Ne	345	2×10^{-2}	138 (368)
pf	20	30	^{42}Ca	9	10^{-8}	116 (304)
		565	^{44}Ca	132	10^{-2}	153 (304)
		3952	^{46}Ca	124	10^{-2}	139 (304)
		12022	^{48}Ca	101	10^{-2}	137 (304)
		17276	^{50}Ca	221	10^{-2}	130 (304)

Number of circuits

$$= N_{terms} \times N_{shots} \times N_{calls}$$

grows as $O(S^4)$
S = num. orbitals

grows as $\frac{1}{\epsilon_E^2}$

depends on optimizer, nucleus

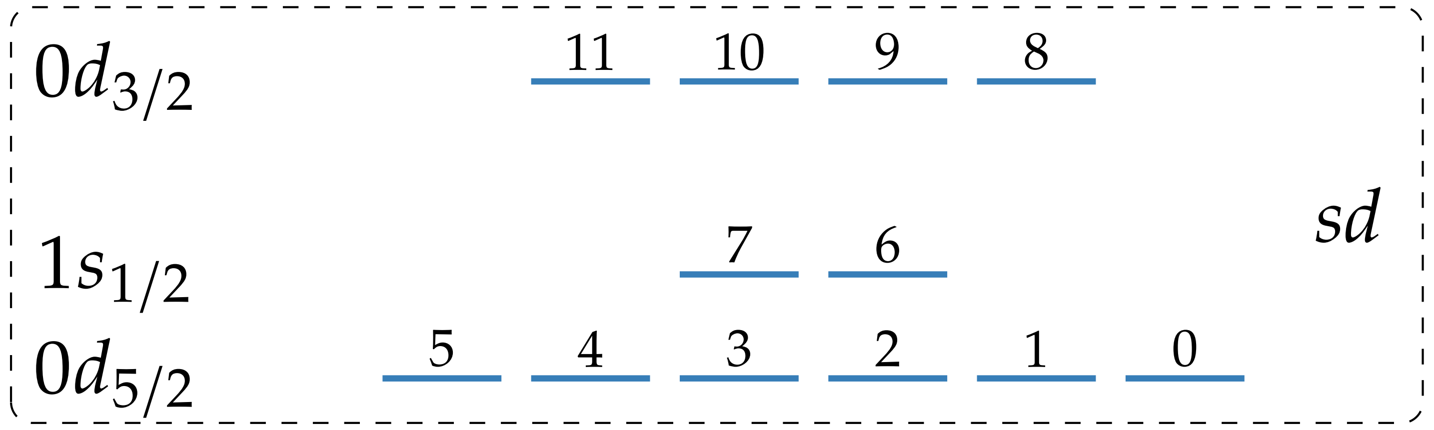
group into terms that self-commute

shell	N_q	N_h	N_{hh}	N_{terms}
p	6	2	10 (9)	13 (12)
	12	4	109 (44)	114 (49)
sd	12	8	203 (86)	212 (95)
	24	16	1389 (518)	1406 (535)
pf	20	20	1507 (570)	1528 (591)
	40	40	10572 (3459)	10613 (3500)

reduces N_{terms} a factor 3

(orbital) entanglement entropies

$$S_A = \rho_A \log_2(\rho_A)$$

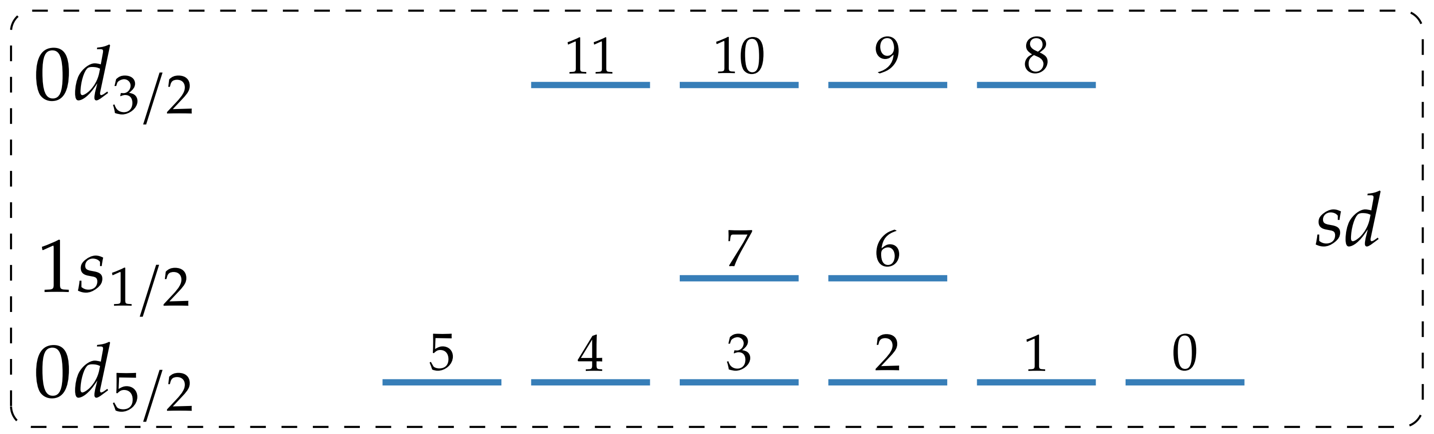


$\rho_A = Tr_B(\rho)$

S

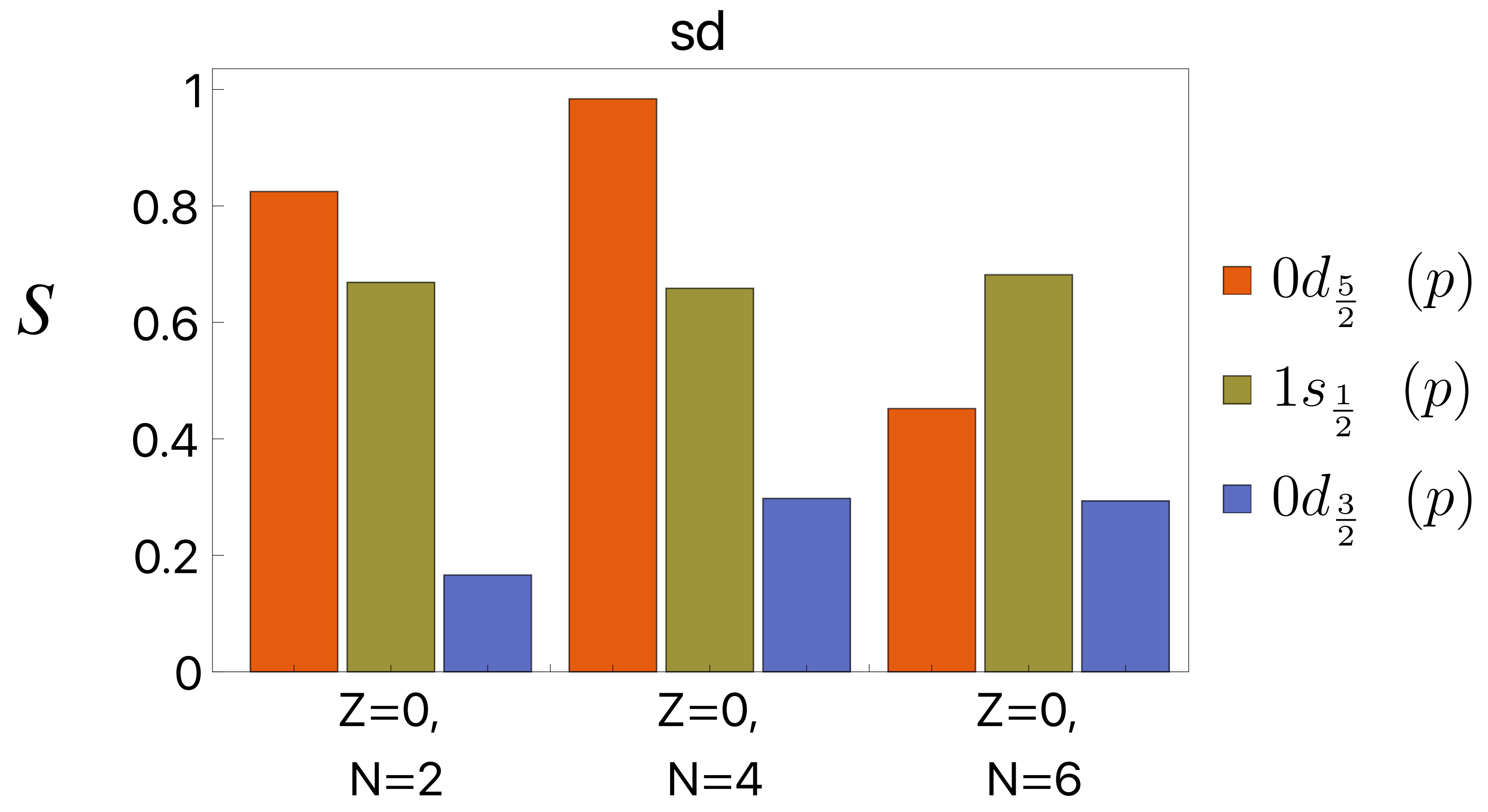
(orbital) entanglement entropies

$$S_A = \rho_A \log_2(\rho_A)$$



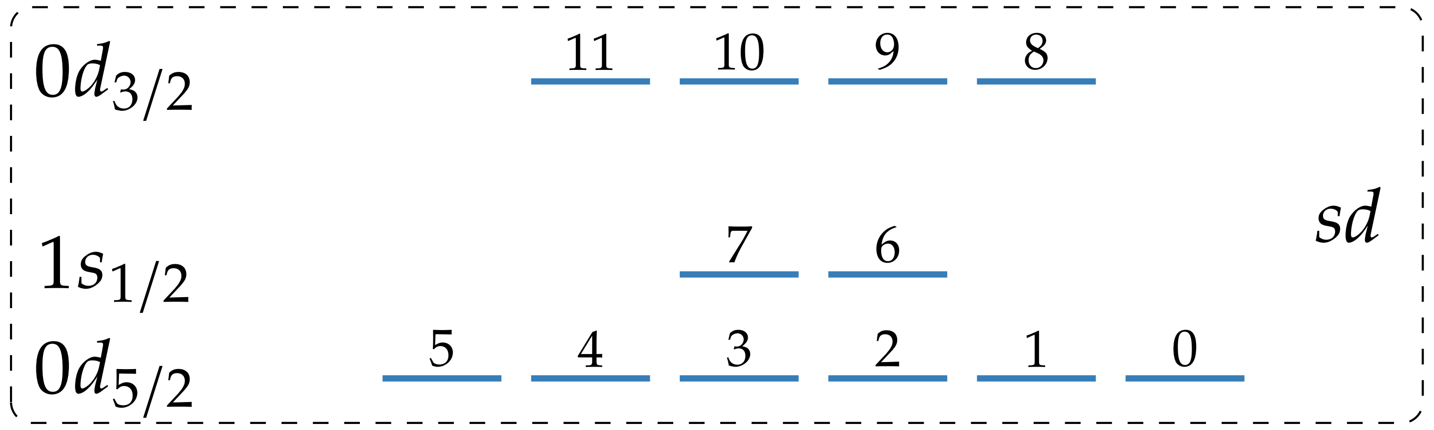
$$\rho_A = Tr_B(\rho)$$

orbital-nucleus



(orbital) entanglement entropies

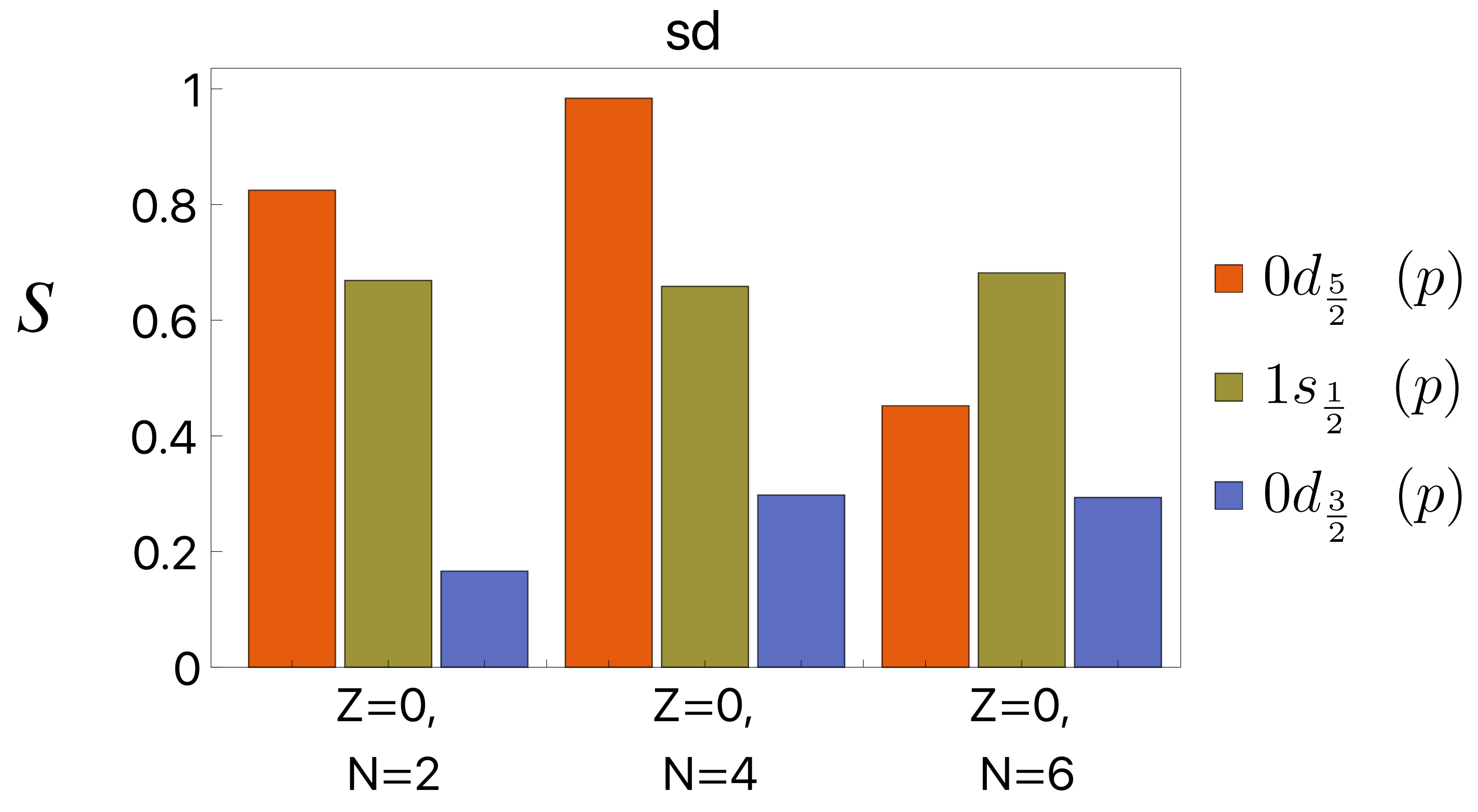
$$S_A = \rho_A \log_2(\rho_A)$$

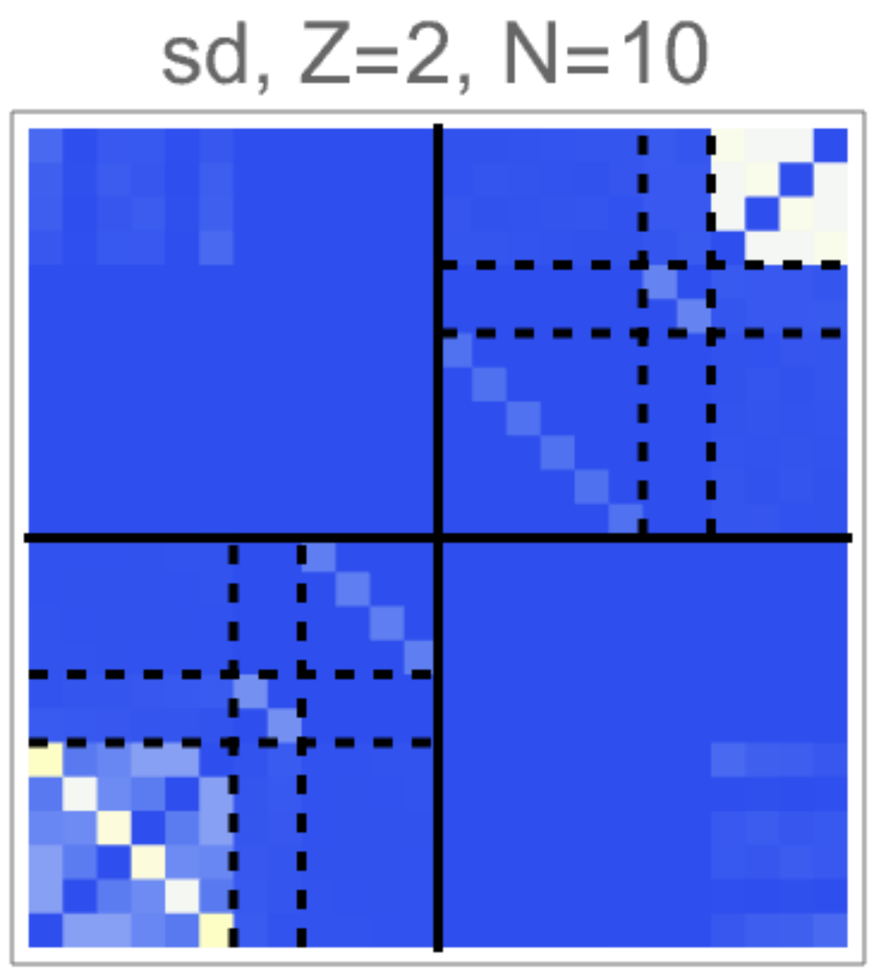


$\rho_A = Tr_B(\rho)$

orbital-nucleus

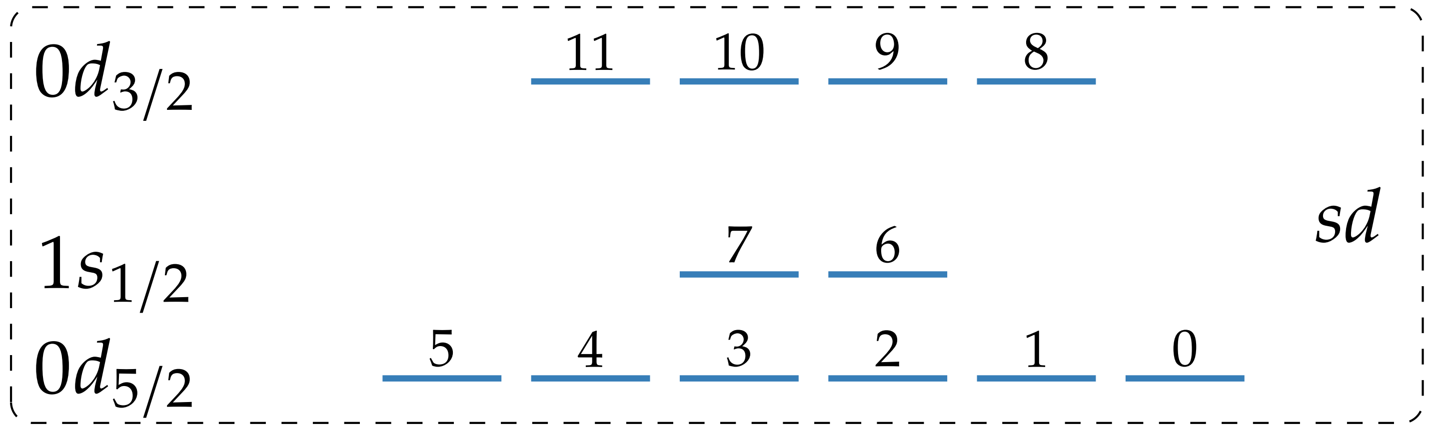
2-orbital $S_i + S_j - S_{ij}$





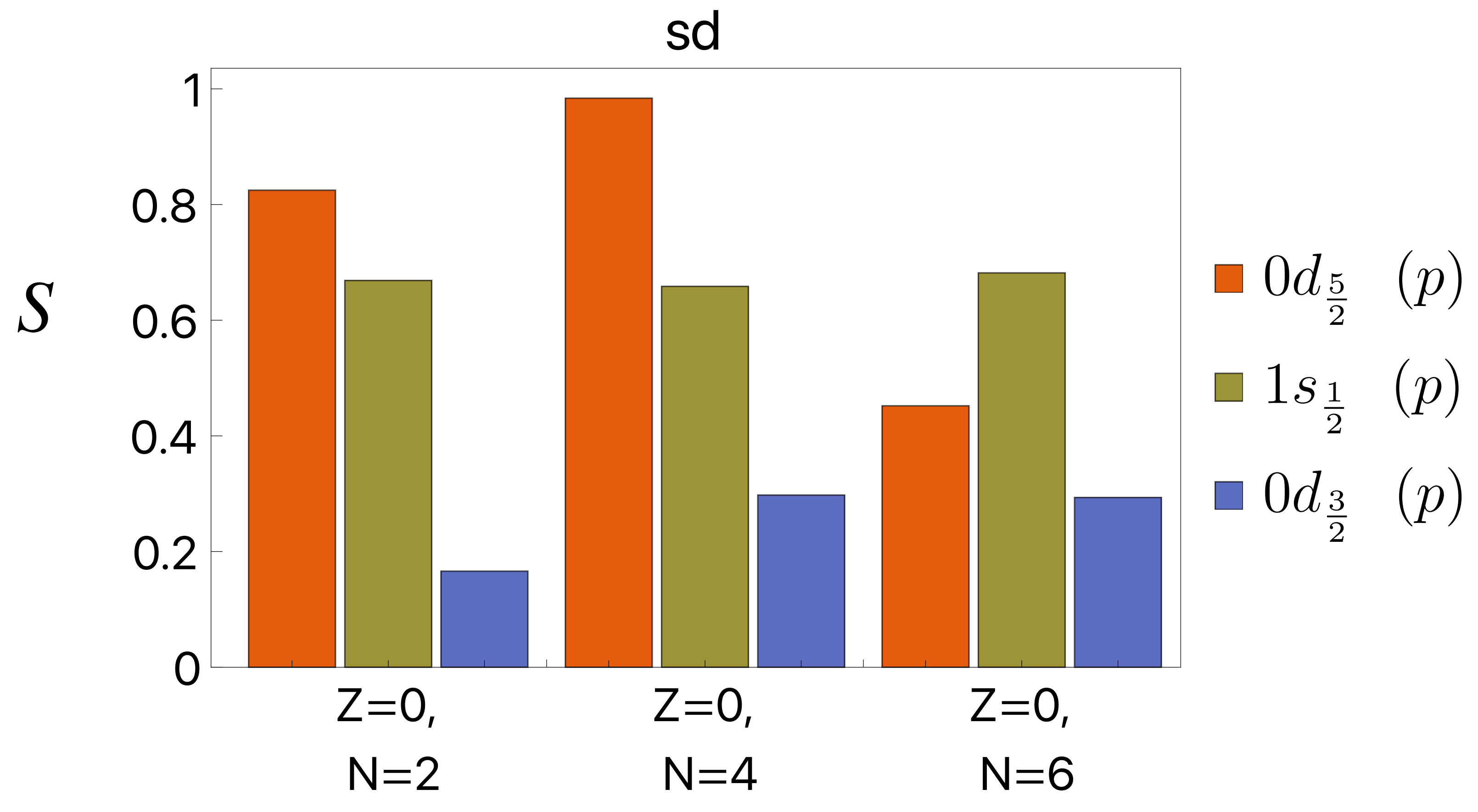
(orbital) entanglement entropies

$$S_A = \rho_A \log_2(\rho_A)$$

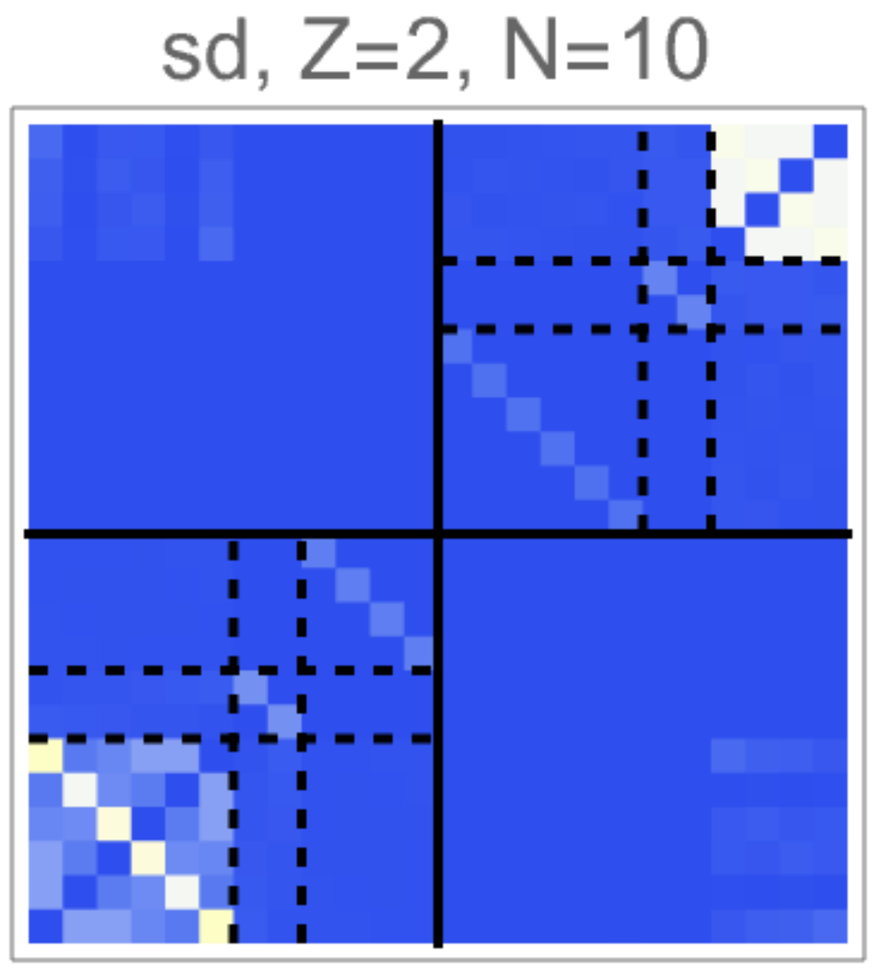


$\rho_A = Tr_B(\rho)$

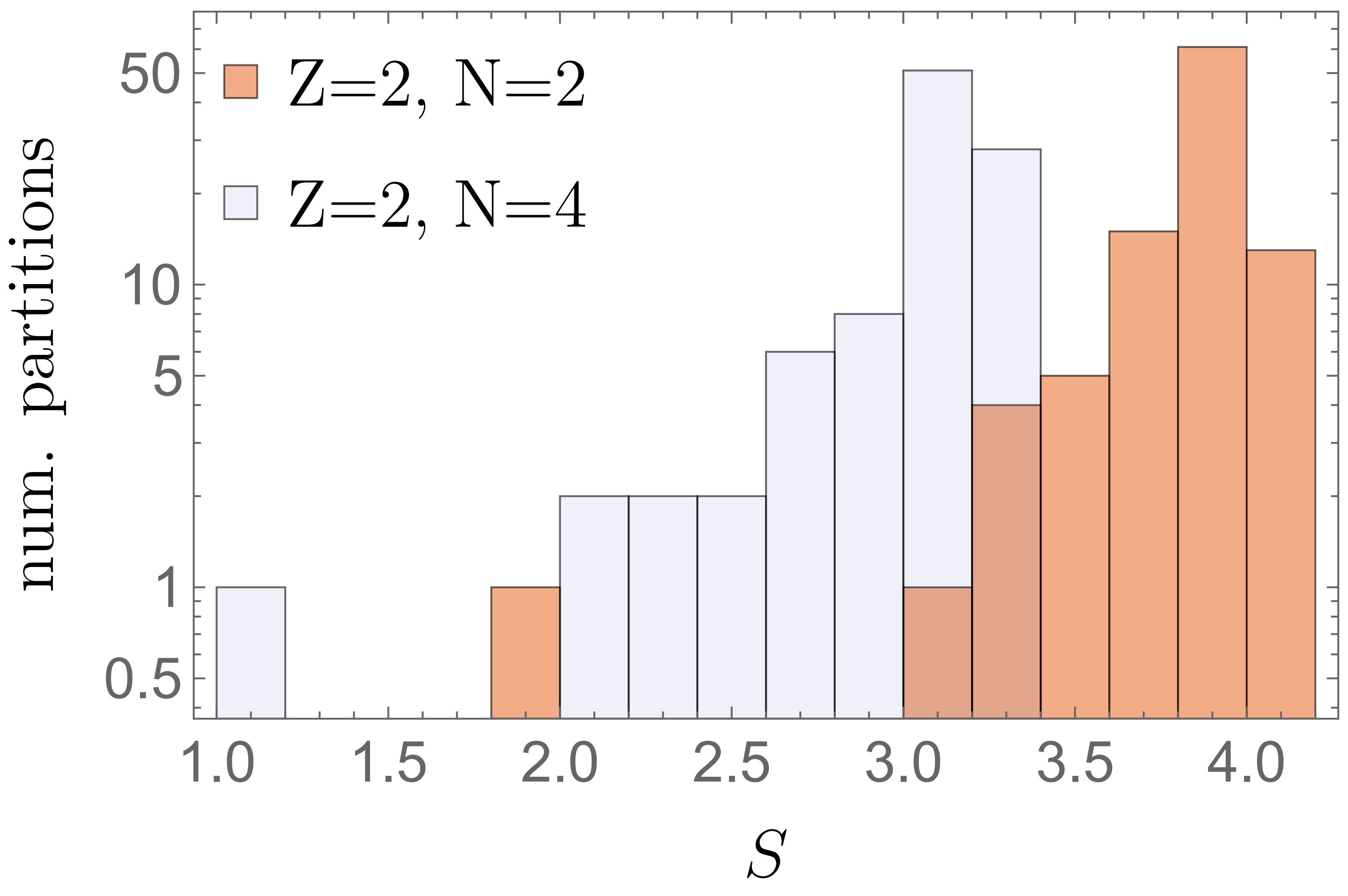
orbital-nucleus



2-orbital $S_i + S_j - S_{ij}$

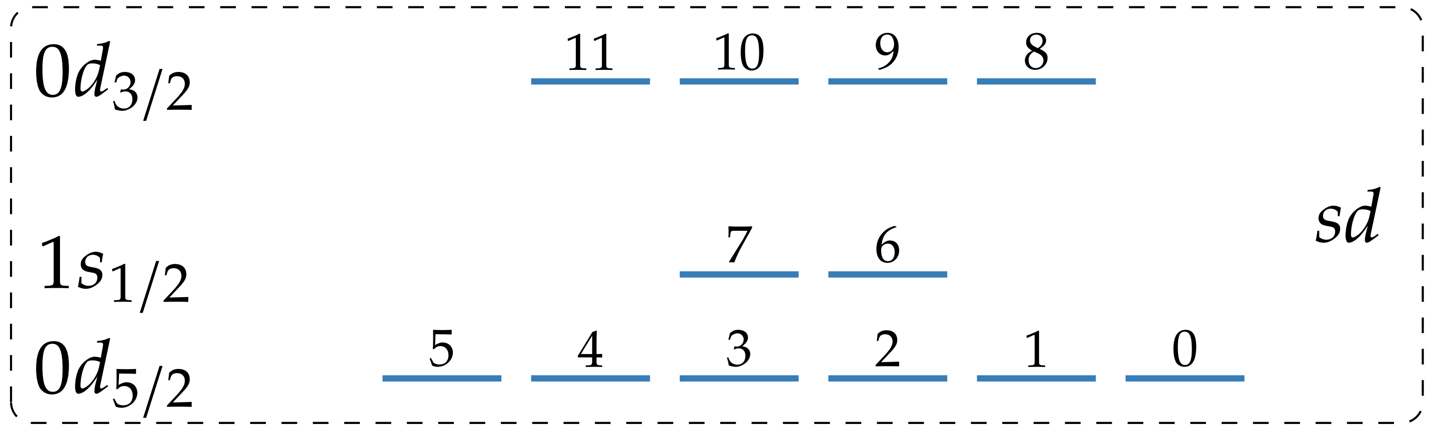


equipartitions



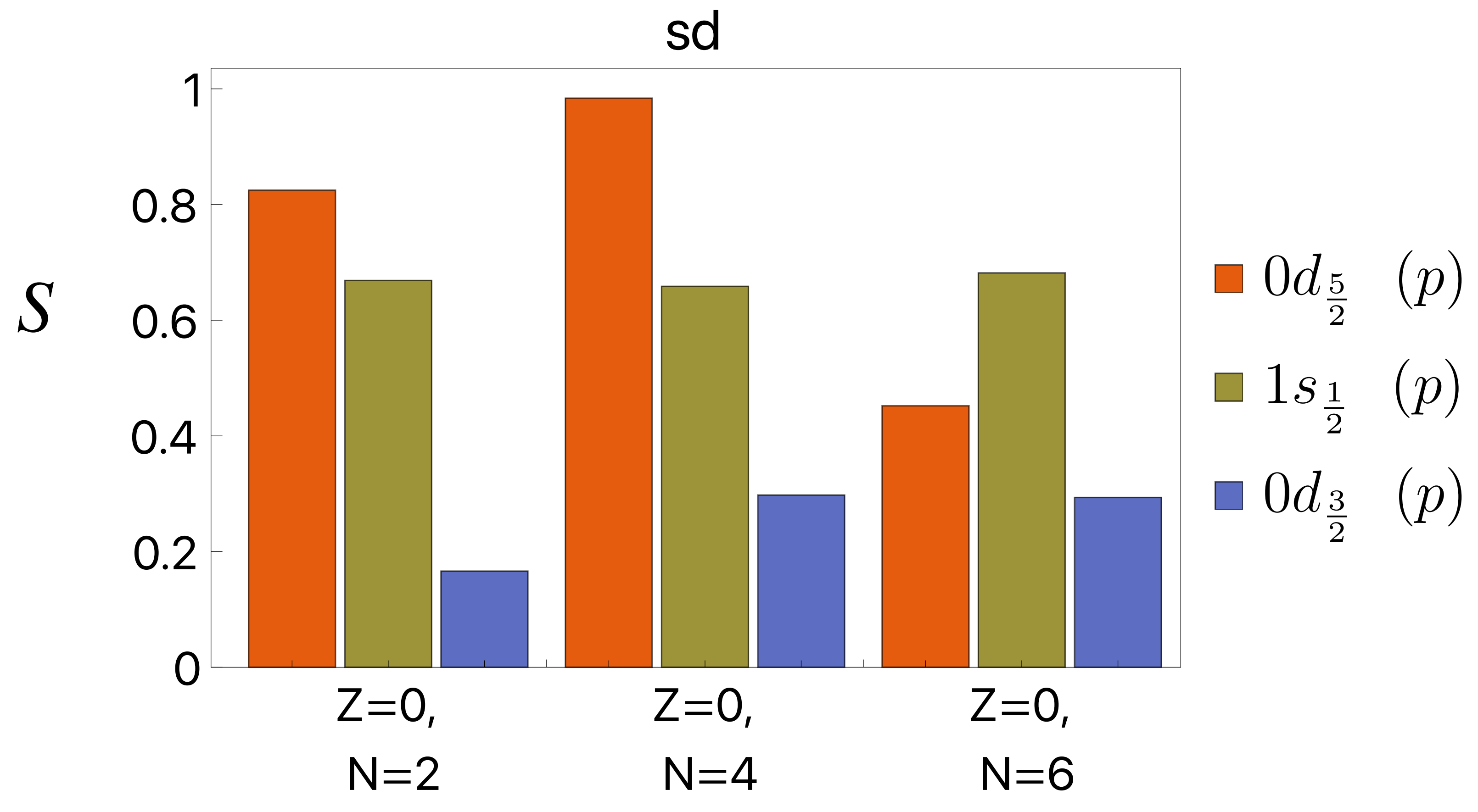
(orbital) entanglement entropies

$$S_A = \rho_A \log_2(\rho_A)$$

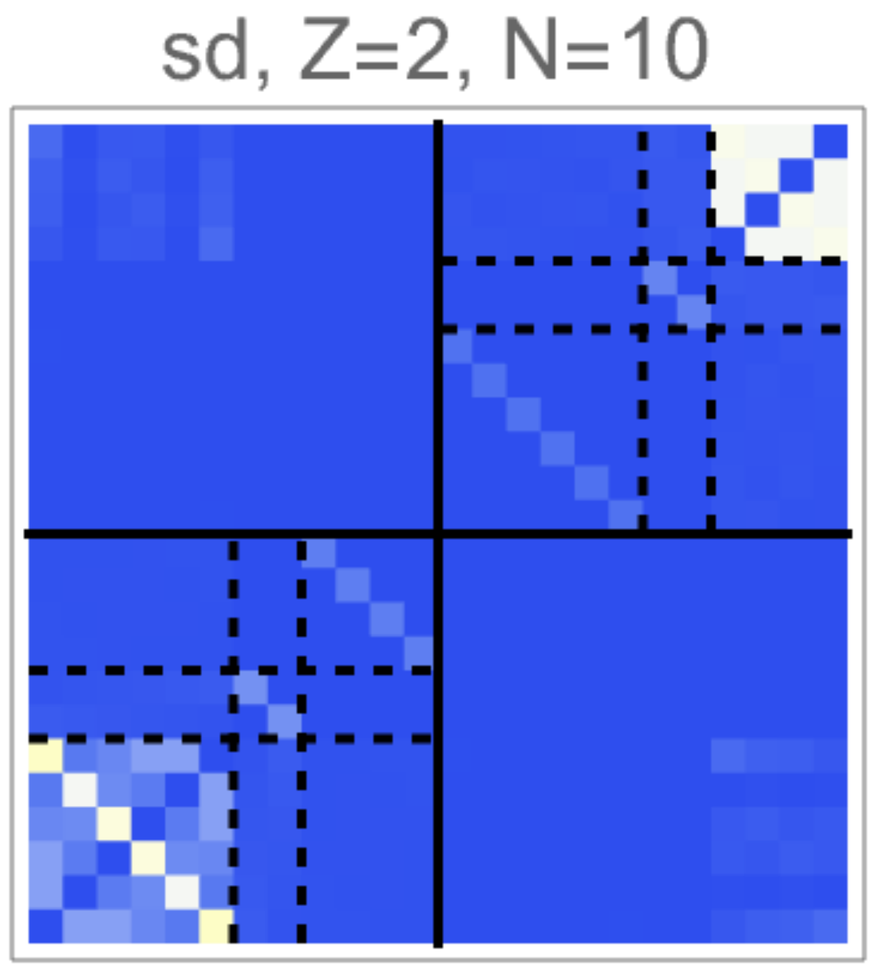


$$\rho_A = Tr_B(\rho)$$

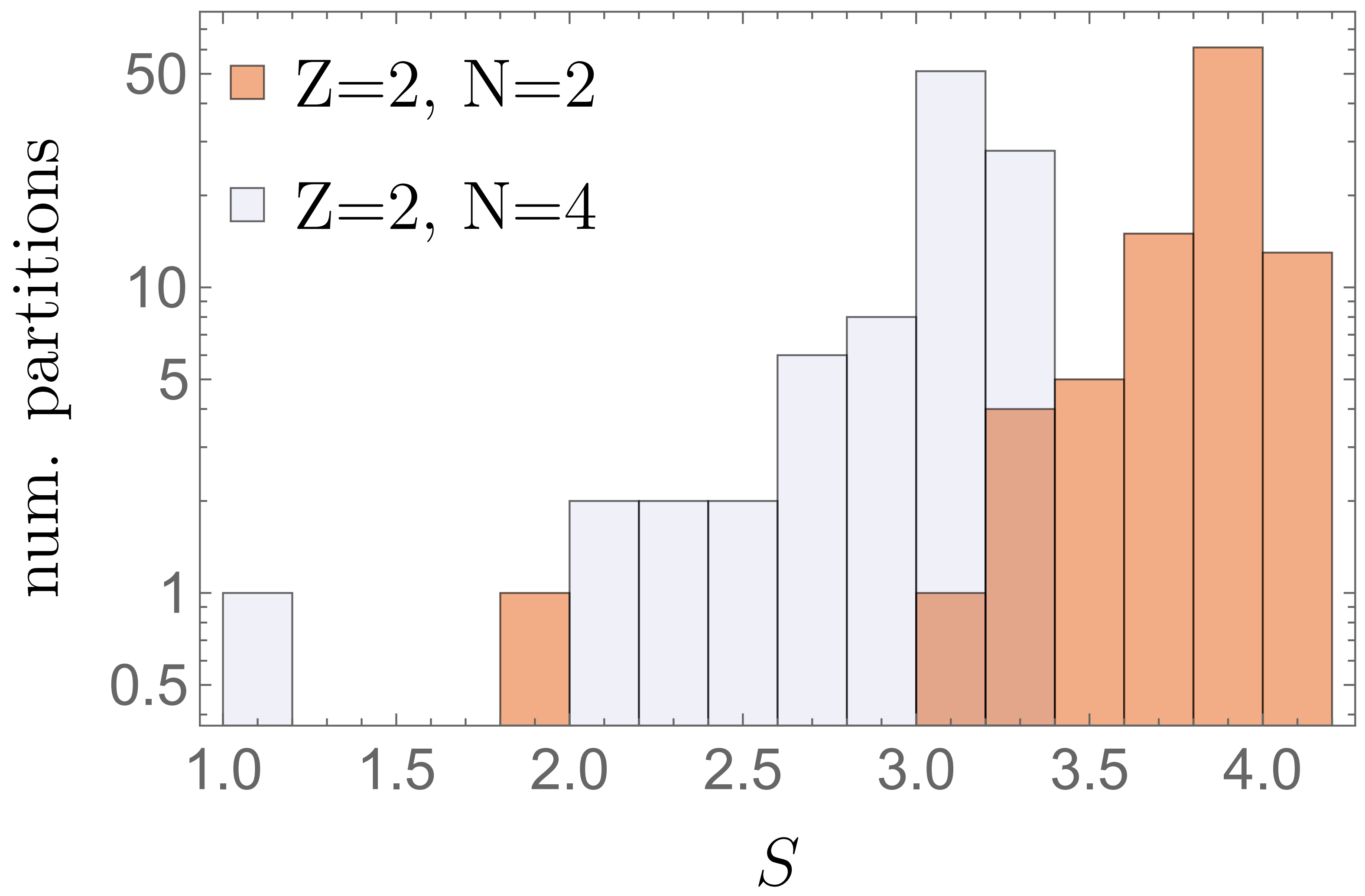
orbital-nucleus



2-orbital $S_i + S_j - S_{ij}$



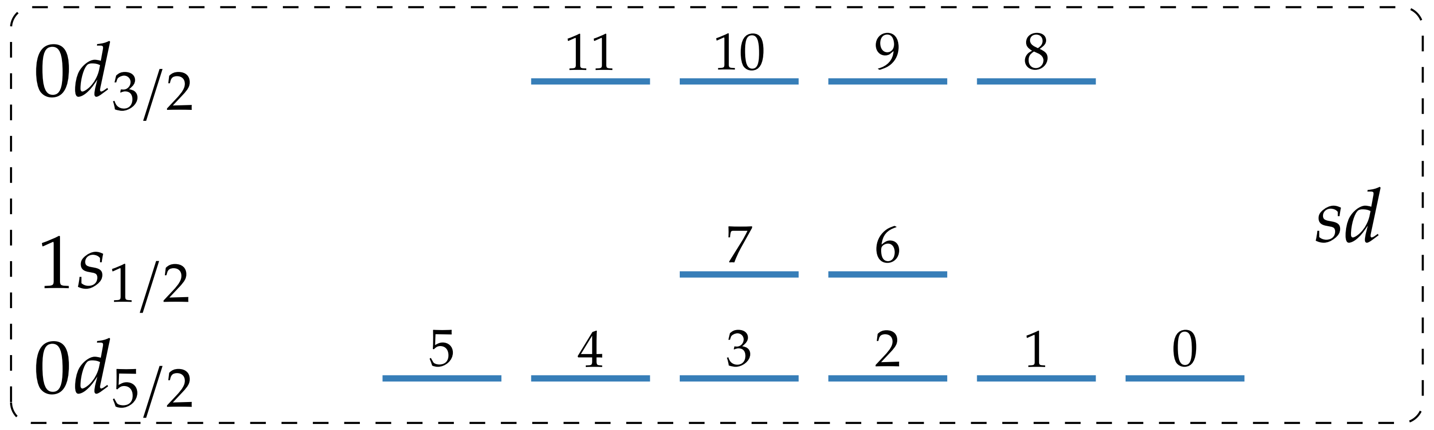
equipartitions



1. Well reproduced with adapt-vqe
2. Shell structure strongly affects entanglement
3. protons and neutrons lowly entangled

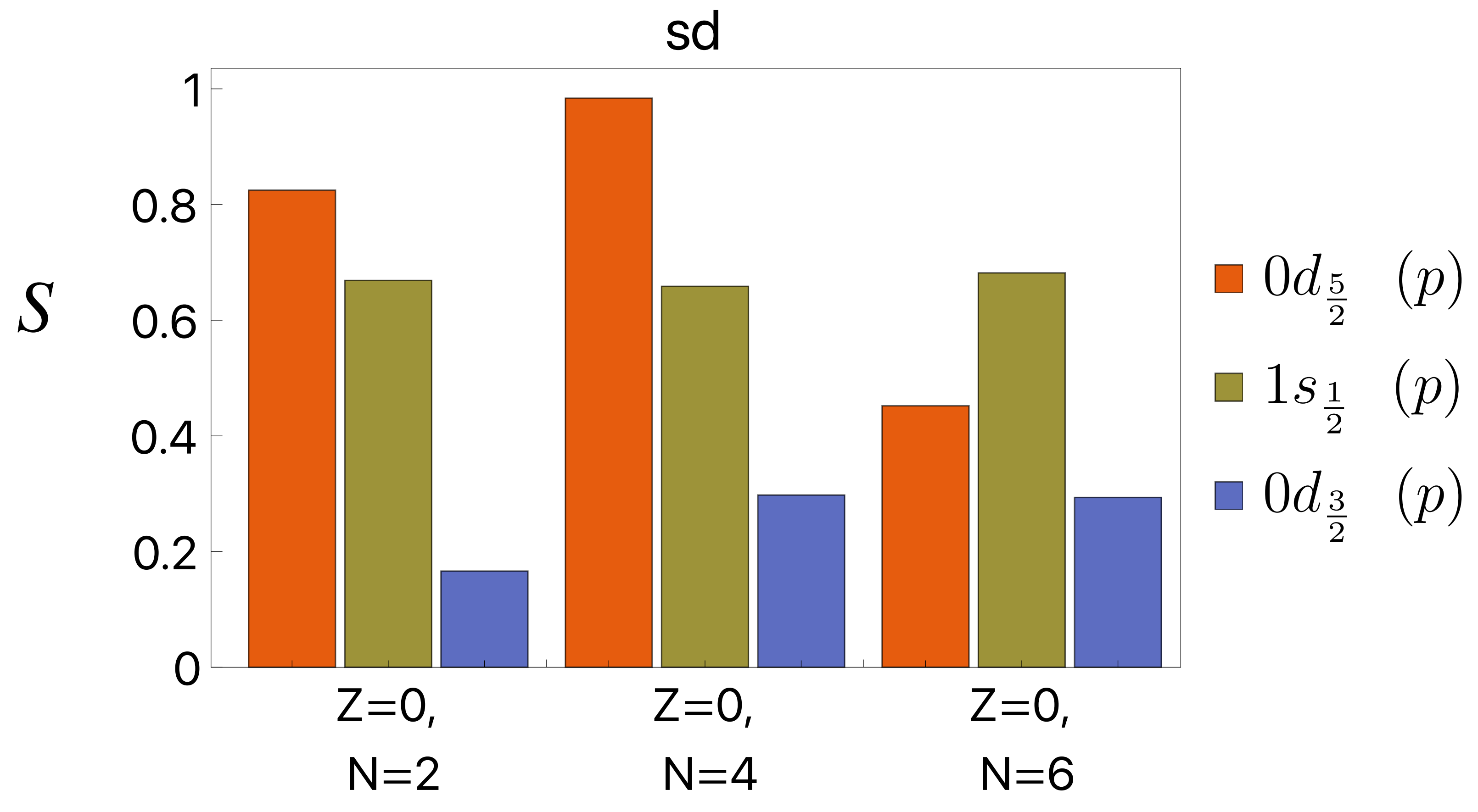
(orbital) entanglement entropies

$$S_A = \rho_A \log_2(\rho_A)$$



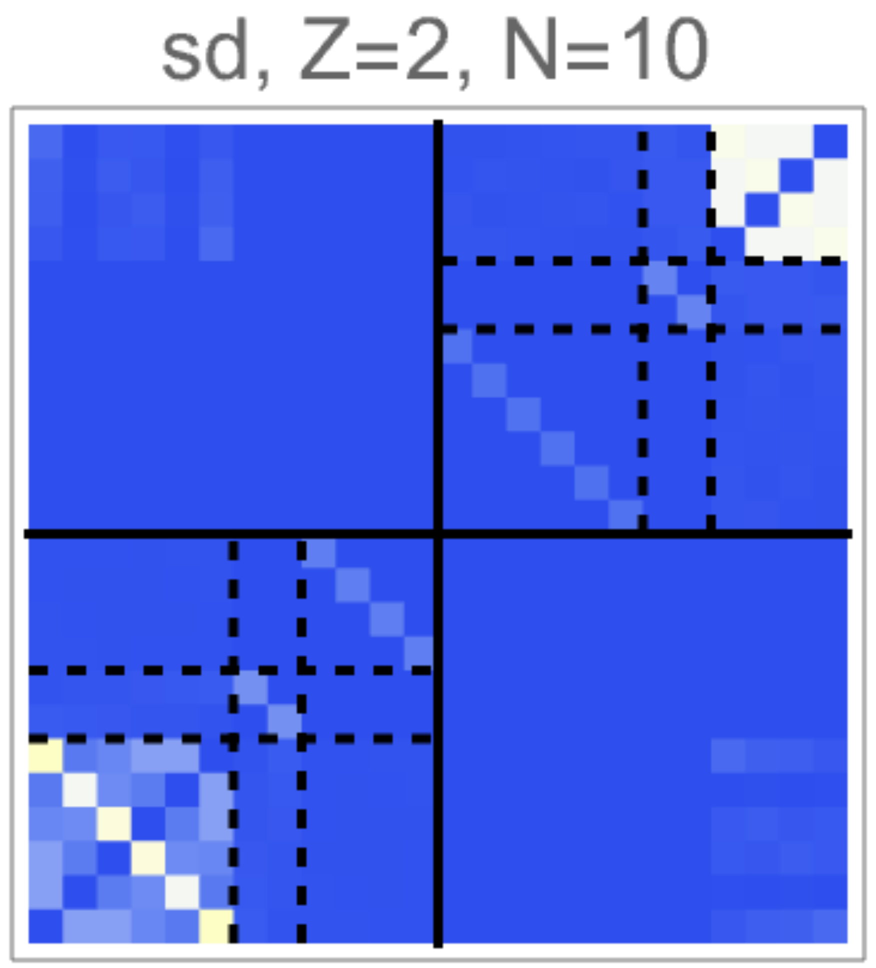
$\rho_A = Tr_B(\rho)$

orbital-nucleus



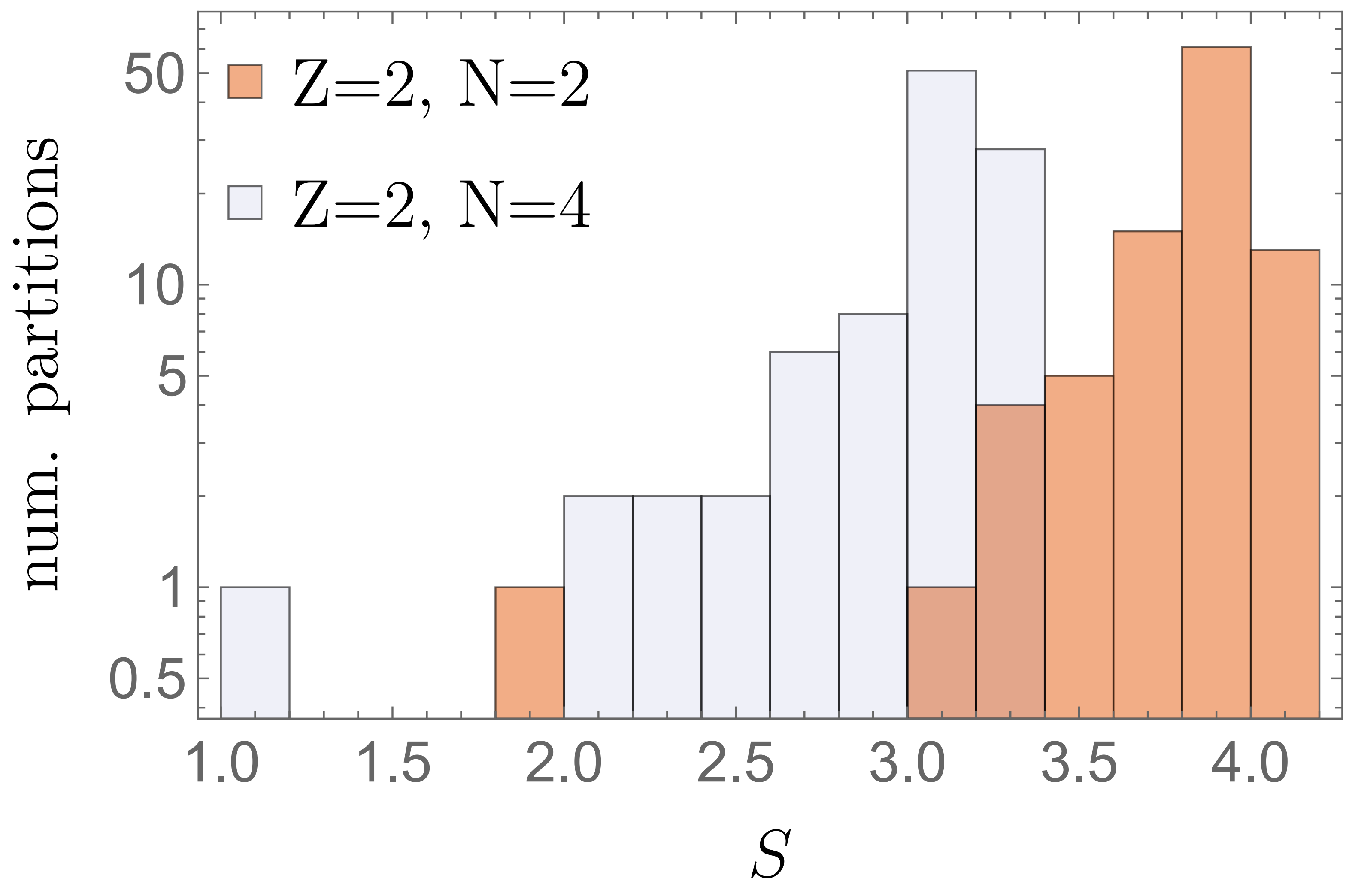
1. Well reproduced with adapt-vqe

2-orbital $S_i + S_j - S_{ij}$



2. Shell structure strongly affects entanglement

equipartitions



3. protons and neutrons lowly entangled

Conclusions

1. Atomic nuclei are good candidates for (digital) quantum simulation:

   ➡ Naturally quantized in shells & orbitals —> easily mapped to qubits

   ➡ Can start with light nuclei and gradually increase to heavier ones

Conclusions

1. Atomic nuclei are good candidates for (digital) quantum simulation:

➡ Naturally quantized in shells & orbitals —> easily mapped to qubits

➡ Can start with light nuclei and gradually increase to heavier ones

2. ADAPT-VQE works well to simulate nuclear ground states

➡ Polynomial scaling with the size of the nuclear shell

➡ No barren plateaus!

➡ Adapts well to quantum hardware

➡ Fidelity and entropies well reproduced

Conclusions

1. Atomic nuclei are good candidates for (digital) quantum simulation:

   ➡ Naturally quantized in shells & orbitals —> easily mapped to qubits

   ➡ Can start with light nuclei and gradually increase to heavier ones

2. ADAPT-VQE works well to simulate nuclear ground states

   ➡ Polynomial scaling with the size of the nuclear shell

   ➡ No barren plateaus!

   ➡ Adapts well to quantum hardware

   ➡ Fidelity and entropies well reproduced

preprint at:

arXiv:2302.03641

other recent work:

quantum simulation for graphene
PRA 106, 052408 (2022)

check Maria Cea’s poster

ultracold gases in digital QC