

# Quantum algorithms for Nuclear Physics III

Dean Lee  
Facility for Rare Isotope Beams  
Michigan State University

Co-ordinated Mini-Lecture Series on  
Quantum Computing and Quantum Information Science  
for Nuclear Physics

Jefferson Laboratory  
March 17, 2020



MICHIGAN STATE  
UNIVERSITY



**NUCLEI**  
Nuclear Computational Low-Energy Initiative  
A SciDAC-4 Project



## Outline

Nuclear dynamics on a quantum chip

Designer pulse control

Two-nucleon system

Shor's algorithm

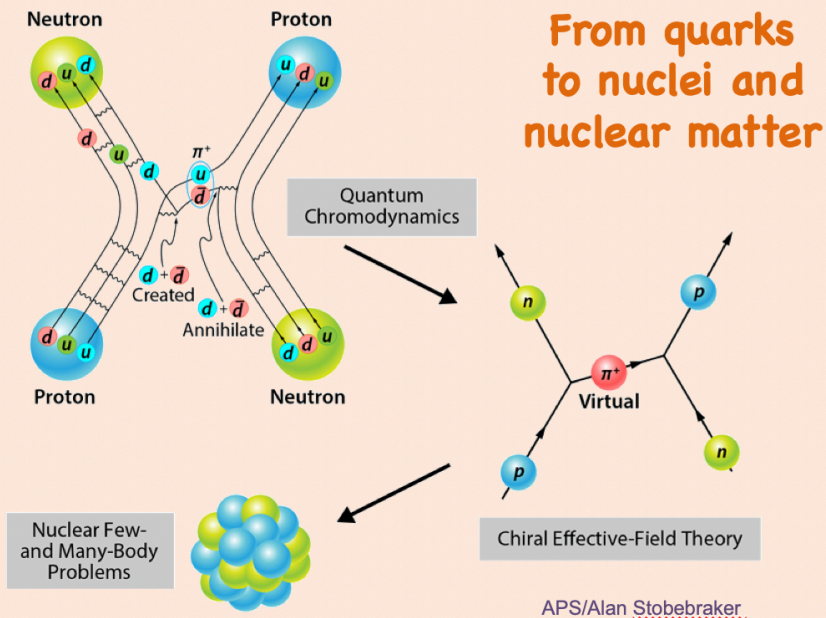
# Nuclear Dynamics on a Quantum Chip

Nov 12, 2019

S. Quaglioni (PI), J. DuBois (co-PI),  
K. Kravvaris, E. Ormand, F. Pederiva, Y. Rosen, K. Wendt, X. Wu



# Goal: Describe entire landscape of nuclear properties in a controlled and unified way grounded in QCD

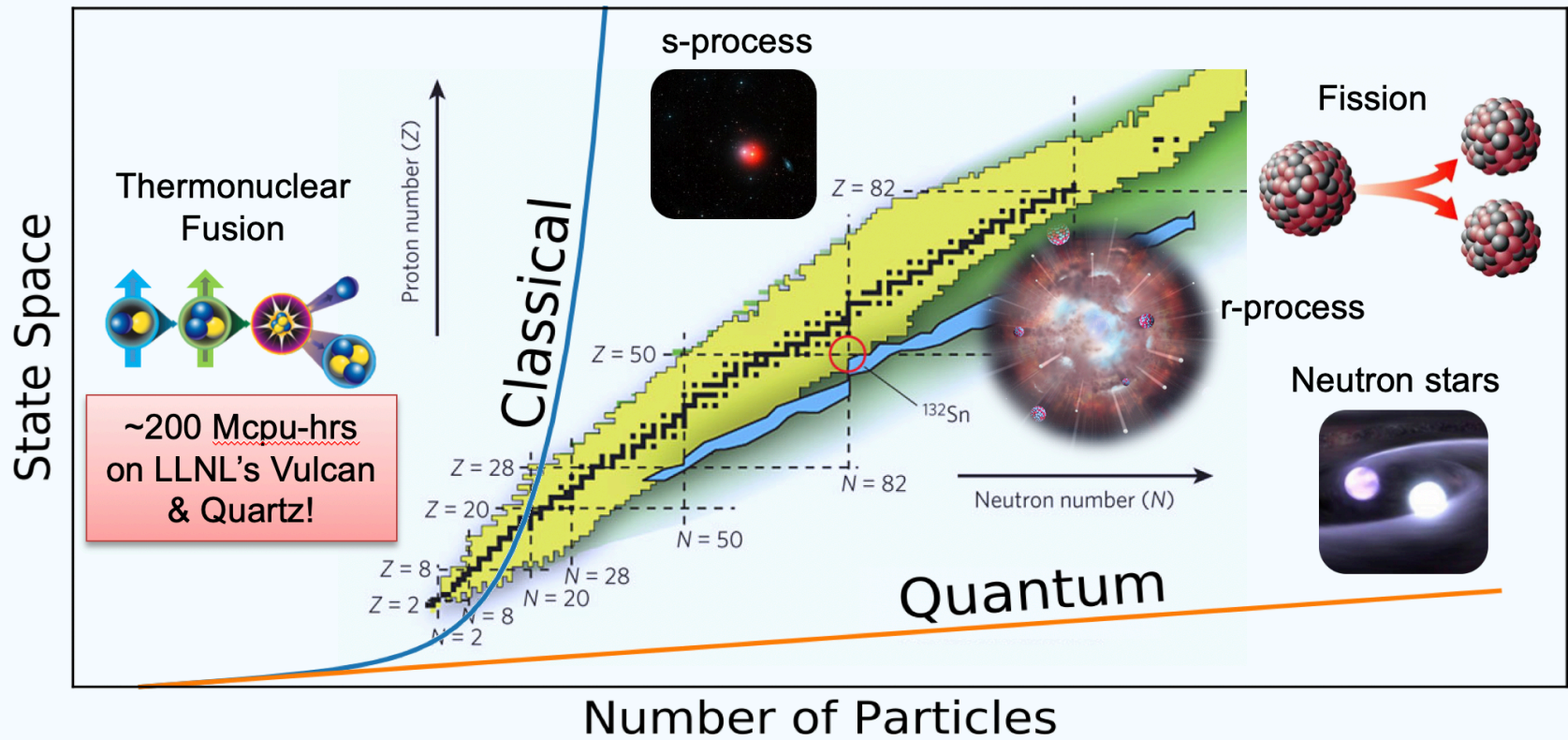


- *Ab initio* many-body calculations
  - A (all active) point-like nucleons
  - Nucleon-nucleon and three-nucleon (NN+3N) interactions guided by Quantum Chromodynamics (QCD)
  - Non relativistic Quantum Mechanics

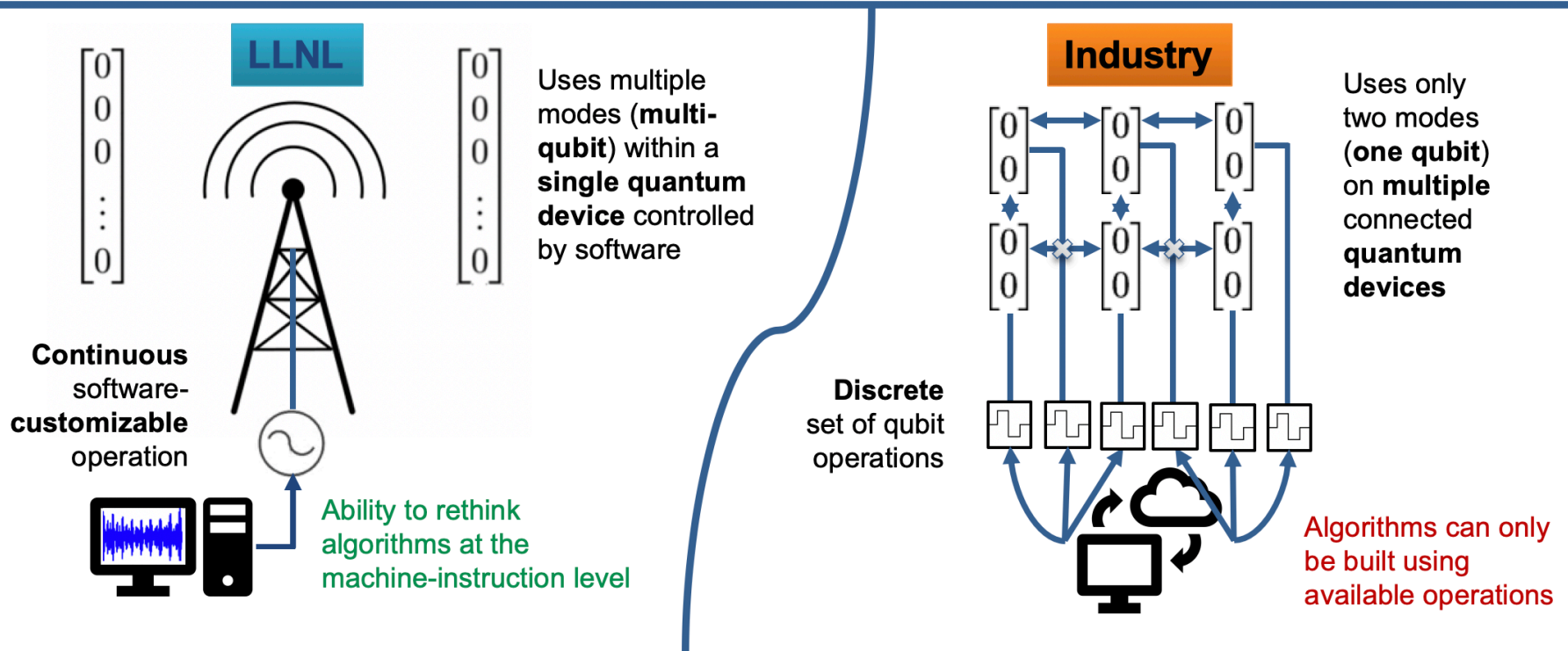
The *ab initio* nuclear many-body problem is extremely complicated and among the most computationally intensive fields of science. Requires efficient theoretical frameworks and large-scale calculations.



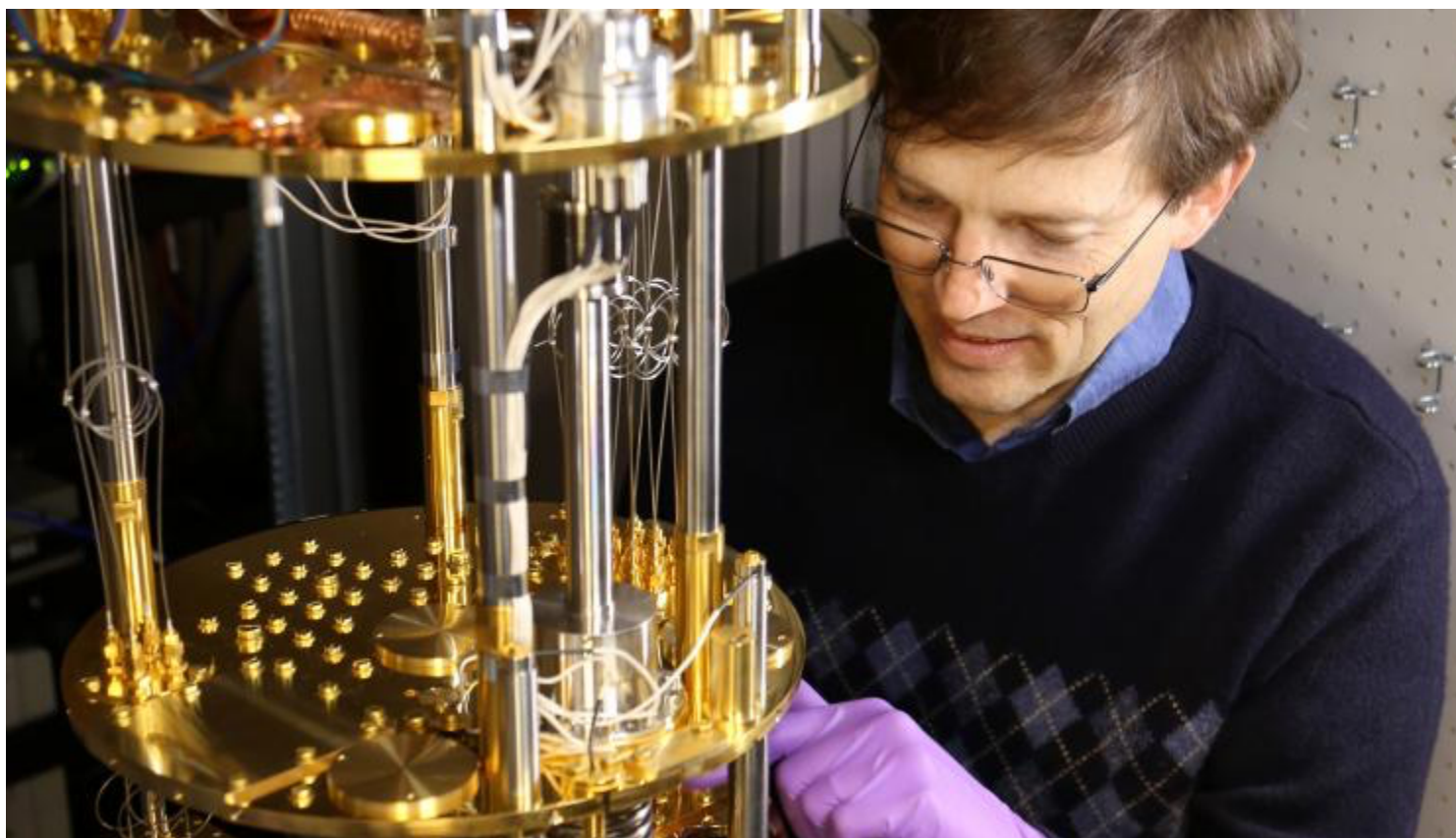
Vast majority of dynamical processes out of reach even with exascale computing. Promise of exact simulations with quantum processors.



# The LLNL testbed differs from superconducting-cavity quantum processing units found in industry in architecture and control



The LLNL testbed adopts a software-extensible quantum computing architecture



Lawrence Livermore National Laboratory physicist Jonathan DuBois, who heads the Lab's Quantum Coherent Device Physics (QCDP) group, examines a prototype quantum computing device designed to solve quantum simulation problems. The device is kept inside a refrigerated vacuum tube (gold-plated to provide solid thermal matching) at temperatures colder than outer space. Photos by Carrie Martin/LLNL

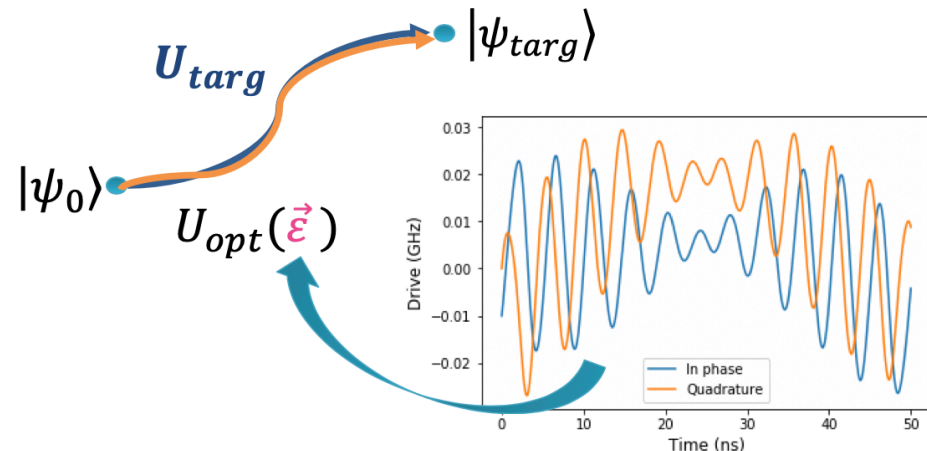
# Our disruptive solution: Single-gate, control model of quantum computing

- A quantum computer performs calculations by manipulating quantum states
- The manipulation occurs by means of unitary operations (**quantum gates**), the quantum analog of Boolean logic operations (AND, OR, NOT, ...)

$$|\psi_{targ}\rangle = U_{targ} |\psi_0\rangle$$

- Our solution is to use a **single gate** to enact the desired operation  $U_{targ}$  by means of a customized pulse

$$U_{targ} \approx U_{opt}(\vec{\epsilon})$$



# Quantum processing unit (QPU) Hamiltonian

$$H_{QPU} = H_d + H_c(t)$$

- Hamiltonian for a 3D transmon coupled to a readout cavity

$$H_d = \hbar\omega_T \hat{a}_T^\dagger \hat{a}_T + \hbar\omega_R \hat{a}_R^\dagger \hat{a}_R - E_J \left[ \cos \hat{\phi}_J + \frac{\hat{\phi}_J^2}{2} \right]$$

- Time-dependent drive for a single-mode transmon (in the frame of the transmon)

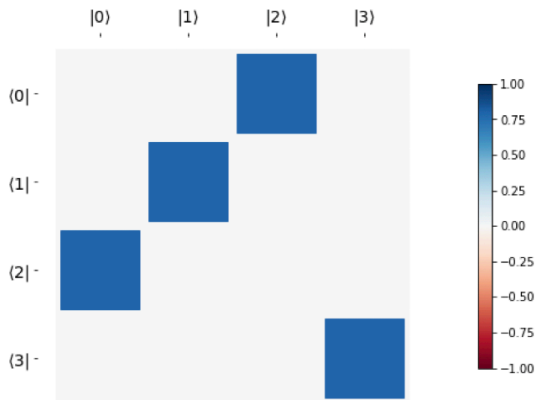
$$H_c(t) = \hbar\varepsilon_I(t)(\hat{a}_T^\dagger + \hat{a}_T) + i\hbar\varepsilon_Q(t)(\hat{a}_T^\dagger - \hat{a}_T)$$



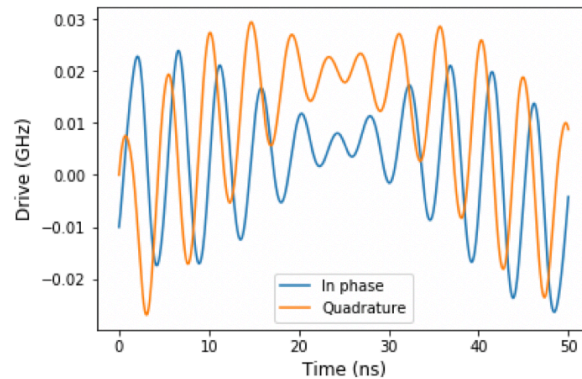
# We use numerical optimization to find a customized drive that implements the target unitary transformation (quantum gate)

$$U_{targ} \cong \mathcal{T} \exp \left\{ -\frac{i}{\hbar} \int_0^\tau H_d + H_c(\tau') d\tau' \right\}$$

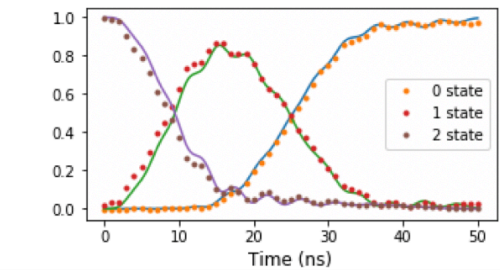
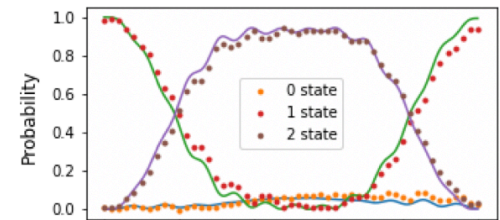
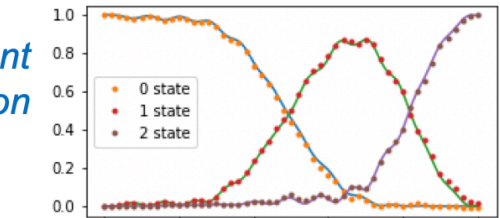
*Experiment vs simulation*



*Target gate: 0 – 2 swap*



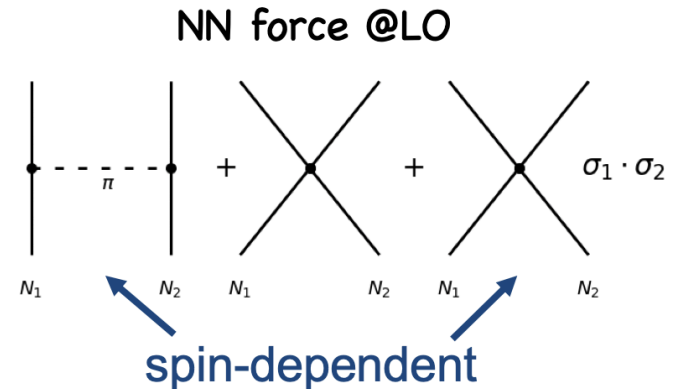
*Solved 50ns Optimized Control Drive of the target gate.*



# Start with a simple, yet non trivial nuclear physics problem

Describe how two nucleons evolve in time under the effect of their mutual interaction

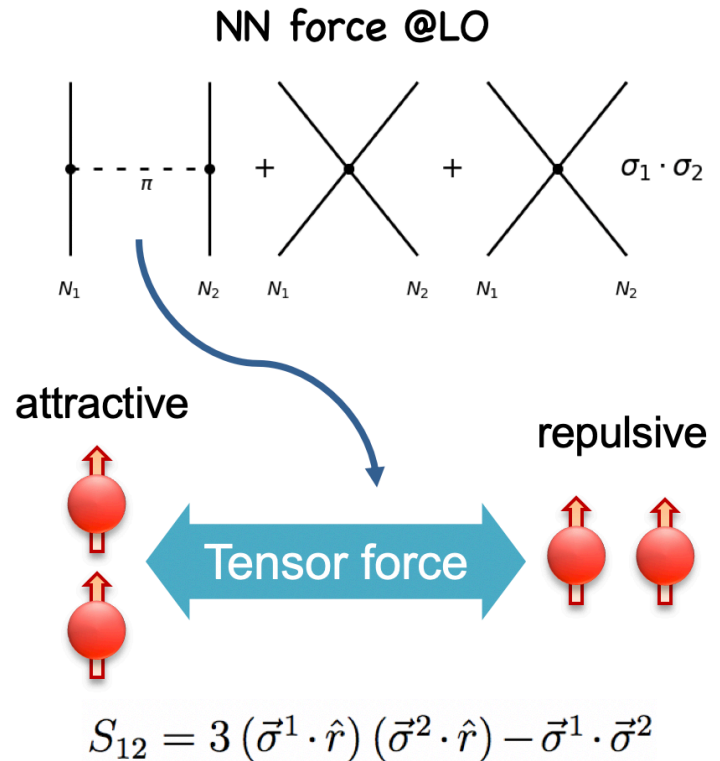
- Interaction at leading-order (LO) of chiral effective field theory (EFT)
- Spin dynamics of 2 neutrons 'frozen in space' (parametric dependence on coordinates)



# Start with a simple, yet non trivial nuclear physics problem

Describe how two nucleons evolve in time under the effect of their mutual interaction

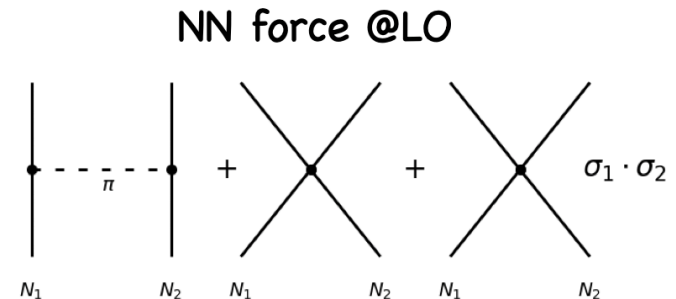
- Interaction at leading-order (LO) of chiral effective field theory (EFT)
- Spin dynamics of 2 neutrons ‘frozen in space’ (parametric dependence on coordinates)
- Non-trivial dependence on the spin state of the neutrons



# Start with a simple, yet non trivial nuclear physics problem

Describe how two nucleons evolve in time under the effect of their mutual interaction

- Interaction at leading-order (LO) of chiral effective field theory (EFT)
- Spin dynamics of 2 neutrons ‘frozen in space’ (parametric dependence on coordinates)
- Quantum-classical co-processing, with spin propagation carried out by a QPU



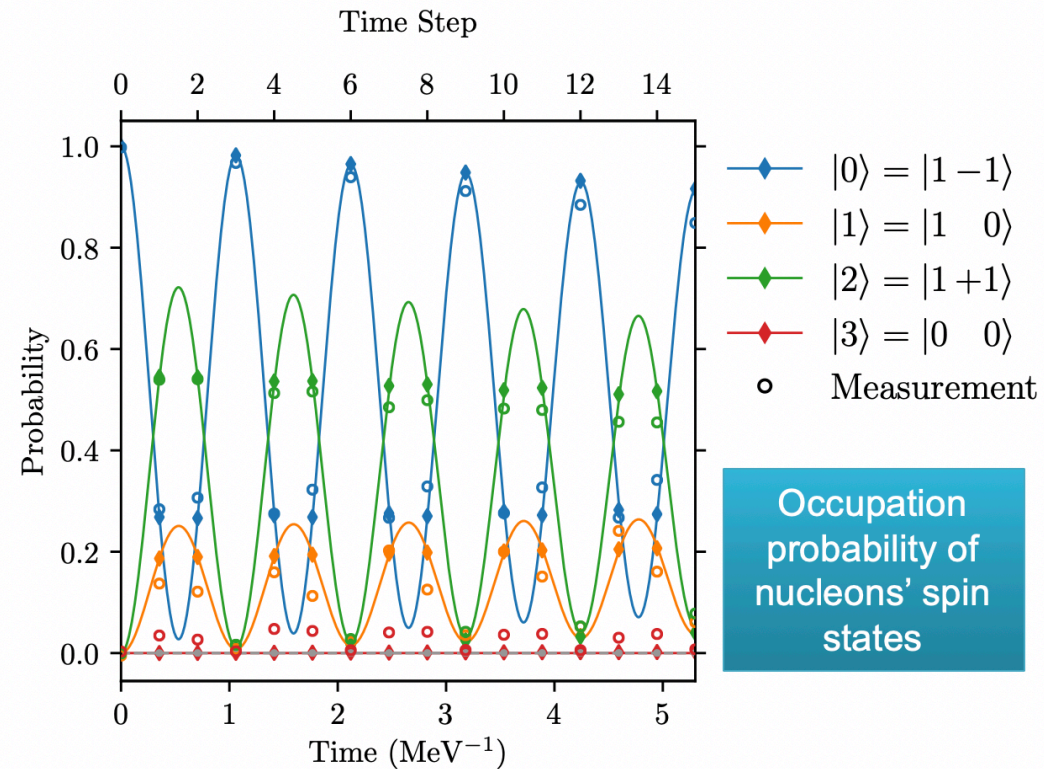
*Unitary time propagator  
(infinitesimal time step)*

$$\exp \left[ -\frac{i}{\hbar} H_{LO} \delta t \right]$$

$$\cong \exp \left[ -\frac{i}{\hbar} H_{SI} \delta t \right] \exp \left[ -\frac{i}{\hbar} V_{SD} \delta t \right]$$

# Demonstration of noise-resilient real-time evolution of two interacting neutrons on the LLNL quantum testbed

- Prepare QPU initial state: e.g.,  $|\downarrow\downarrow\rangle$
- Perform time evolution by dividing into small time steps: repeat same gate at each time step
- Due to the nucleons' interaction, with time the QPU evolves into an entangled superposition of the 3 states
- Measure time-dependent probability with which QPU occupies each spin state





# We use numerical optimization to find a customized drive that implements the desired two-neutron spin short-time propagator

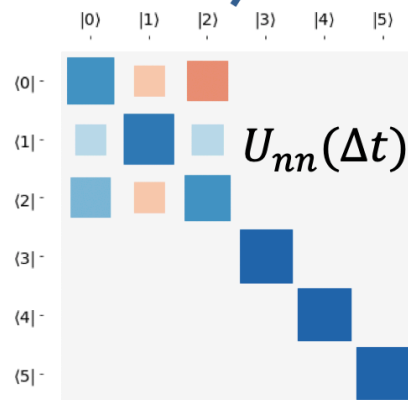
- Nuclear interaction entangles  $S = 1$  states:

$$|\uparrow\uparrow\rangle \rightarrow |0\rangle$$

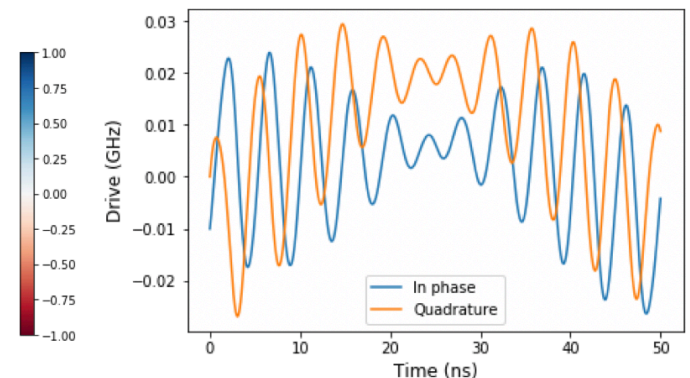
$$\left| \frac{|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle}{\sqrt{2}} \right\rangle \rightarrow |1\rangle$$

$$|\downarrow\downarrow\rangle \rightarrow |2\rangle$$

$$\exp\left[-\frac{i}{\hbar} V_{SD} \Delta t\right] \cong \mathcal{T} \exp\left\{-\frac{i}{\hbar} \int_0^\tau H_d + H_c(\tau') d\tau'\right\}$$



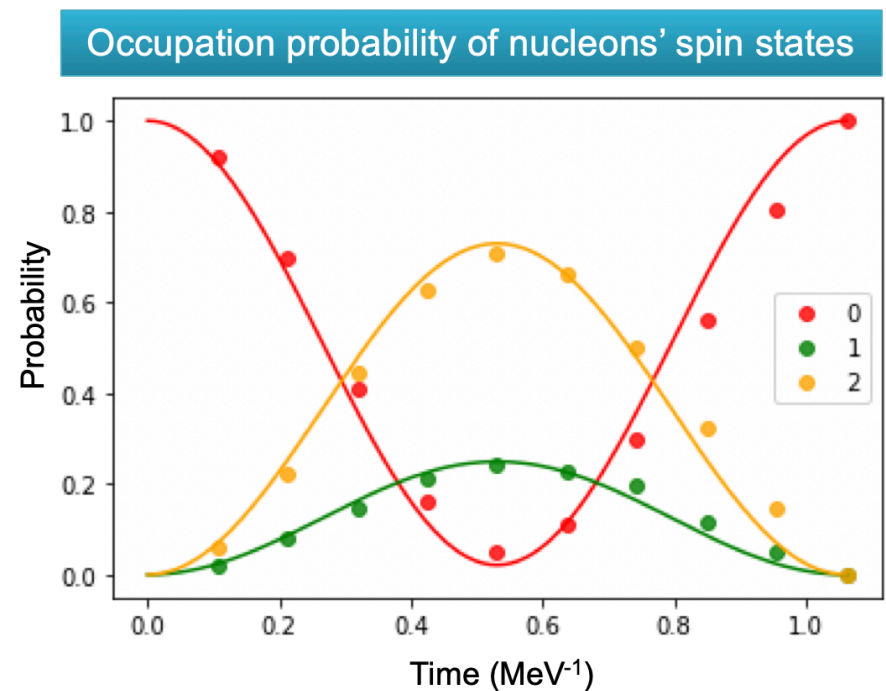
Target gate: two-neutron spin dynamics



Solved Optimized Control Drive.

# Demonstration of noise-resilient real-time evolution of two interacting neutrons on the LLNL quantum testbed

- Prepare QPU initial state: e.g.,  $|\uparrow\uparrow\rangle$
- Perform time evolution by dividing into small time steps: customized gate at each time step
- Due to the nucleons' interaction, with time the QPU evolves into an entangled superposition of the 3 states
- Measure time-dependent probability with which QPU occupies each spin state



## Euclidean algorithm

$$\gcd(A, B) = ?$$

$$A = BQ_1 + R_1$$

$$B = R_1Q_2 + R_2$$

$$R_1 = R_2Q_3 + R_3$$

•  
•  
•

$$R_{M-2} = R_{M-1}Q_N + R_M$$

$$R_M = \gcd(A, B)$$

## Shor's algorithm

### Step 1

Determine if the number  $N$  is a prime, an even number, or an integer power of a prime number. If yes to any of these, we do not need Shor's algorithm. There are efficient classical methods for determining if an integer  $N$  belongs to one of the above categories. This step would be performed on a classical computer.

## Step 2

Pick an integer  $q$  which is the power of 2 such that

$$N^2 \leq q < 2N^2$$

This step would be done on a classical computer.



### Step 3

Pick a random integer  $x$  that is relatively prime to  $N$ . There are efficient classical methods for picking such an  $x$ . This step would be done on a classical computer.

## Step 4

Create a quantum register and partition it into two parts, register 1 and register 2.

Register 1 must have enough qubits to represent integers as large as  $q - 1$ . Register 2 must have enough qubits to represent integers as large as  $N - 1$ . The calculations for how many qubits are needed would be done on a classical computer.

## Step 5

Load register 1 with an equally weighted superposition of states labelled by all integers from 0 to  $q - 1$ . Load register 2 in the zero state. This operation would be performed by the quantum computer. The total state of the quantum memory register at this point is

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, 0\rangle$$

## Step 6

Now apply the transformation  $x^a \bmod N$  to each number stored in register 1 and store the result in register 2. This is the most difficult step of Shor's algorithm to implement, and some have argued that the efficient implementation of modular exponentiation might be an unresolved gap in the algorithm. The state of the quantum memory register at this point is

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, x^a \bmod N\rangle$$

## Step 7

Measure register 2, and observe some value  $k$ . This has the side effect of collapsing register 1 into an equal superposition of each value  $a$  between 0 and  $q - 1$  such that

$$x^a \bmod N = k$$

Let  $A$  be set of all such values for  $a$ . The state of the quantum memory register after this step is

$$\frac{1}{\sqrt{\|A\|}} \sum_{a' \in A} |a', k\rangle$$

The integers in set  $A$  will be spaced apart in multiples of  $r$ , where  $r$  is the multiplicative order of  $x$ . This is the smallest positive number such that

$$x^r \bmod N = 1$$



$$\text{QFT}(|a\rangle) = \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} |c\rangle e^{2\pi i a c / q}$$

## Step 8

Now apply the quantum Fourier transform to register 1. The state in the quantum memory register is now

$$\frac{1}{\sqrt{q}\sqrt{\|A\|}} \sum_{a' \in A} \sum_{c=0}^{q-1} |c, k\rangle e^{2\pi i a' c / q}$$

## Step 9

Measure the state of register 1, call this value  $m$ , this integer  $m$  has a very high probability of being a multiple of  $q/r$ , where  $r$  is the desired period. This step is performed by the quantum computer.

## Step 10

Take the value  $m$ , and on a classical computer do some post processing to determine  $r$  based on knowledge of  $m$  and  $q$ . There are efficient ways to do this post processing on a classical computer.

## Step 11

Once you have attained  $r$  and it is even, then the prime factors in the factorization  $N = pq$  can be determined by taking  $p = \gcd(x^{r/2} + 1, N)$  and  $q = \gcd(x^{r/2} - 1, N)$  or vice versa. The gcd can be computed using the Euclidean algorithm.

This assumes that  $r$  is even. If that is not the case then go back to step 3 and pick another random  $x$  that is relative prime to  $N$ . This final step is done on a classical computer.

## Recap of lecture

Nuclear dynamics on a quantum chip

Designer pulse control

Two-nucleon system

Shor's algorithm