

# Lie algebraic perspectives on Hamiltonian evolution

Alexander (Lex) Kemper

Department of Physics  
North Carolina State University  
<https://go.ncsu.edu/kemper-lab>



RPMBT XXI  
09/12/2022



*Electronic version of  
this talk*

With:

Efekan Kökcü, Thomas Steckmann (NCSU)

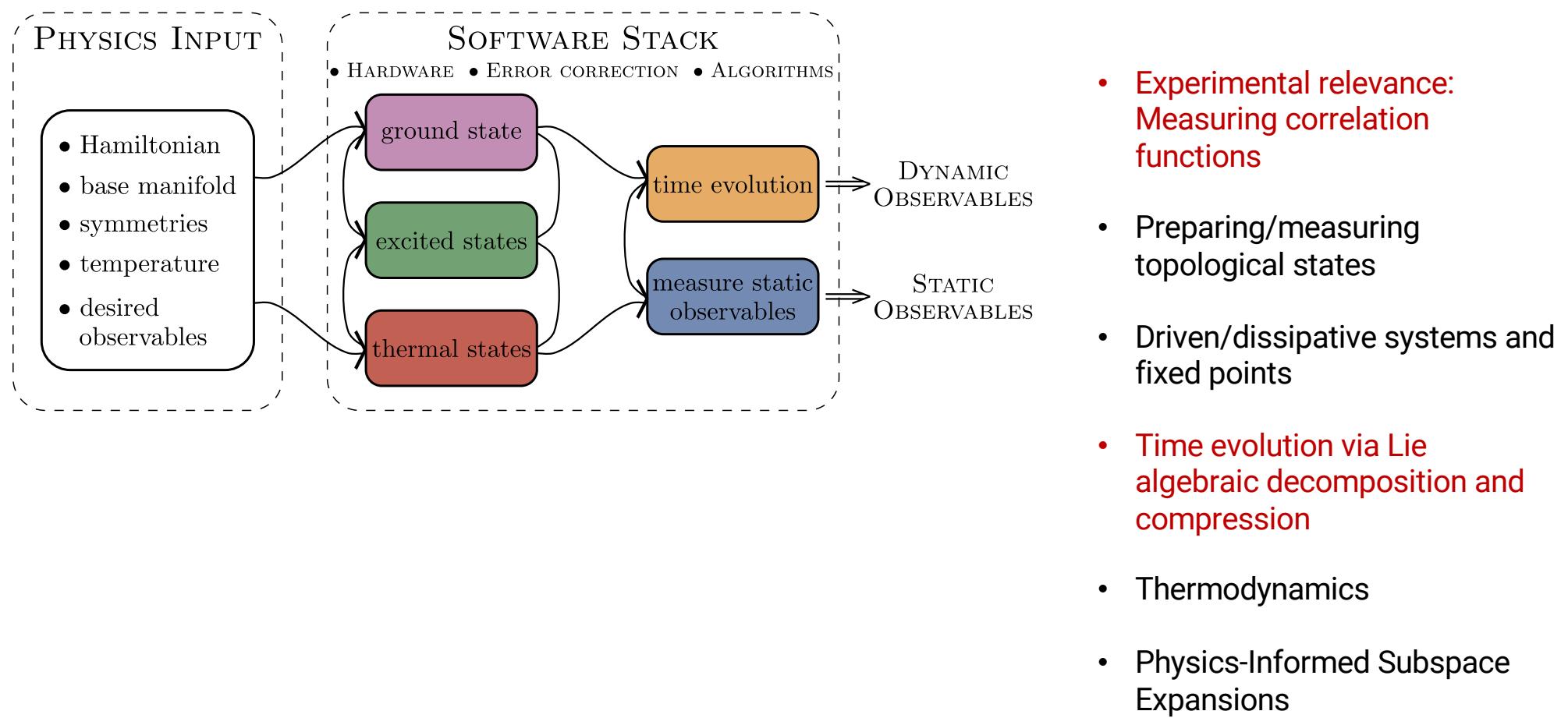
Eugene Dumitrescu & Yan Wang (ORNL)

Daan Camps, Roel van Beeumen, Lindsay Bassman, Bert de Jong (LBNL)

Jim Freericks (Georgetown)



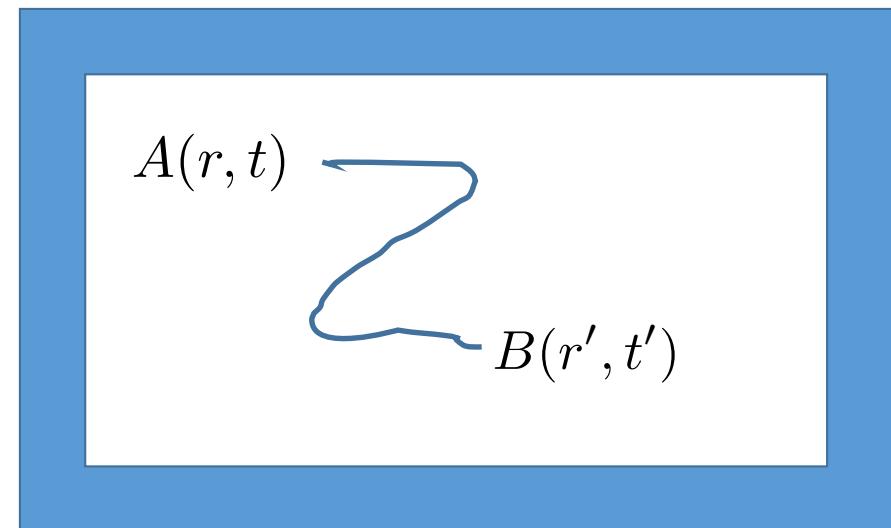
# Quantum Matter meets Quantum Computing



## Low-energy excitations: correlation functions

$$\langle A(r, t)B(r', t') \rangle$$

*Given some (observable) operator B at (r',t'), what is the likelihood of some (observable) operator A at (r,t)?*



Conductivity

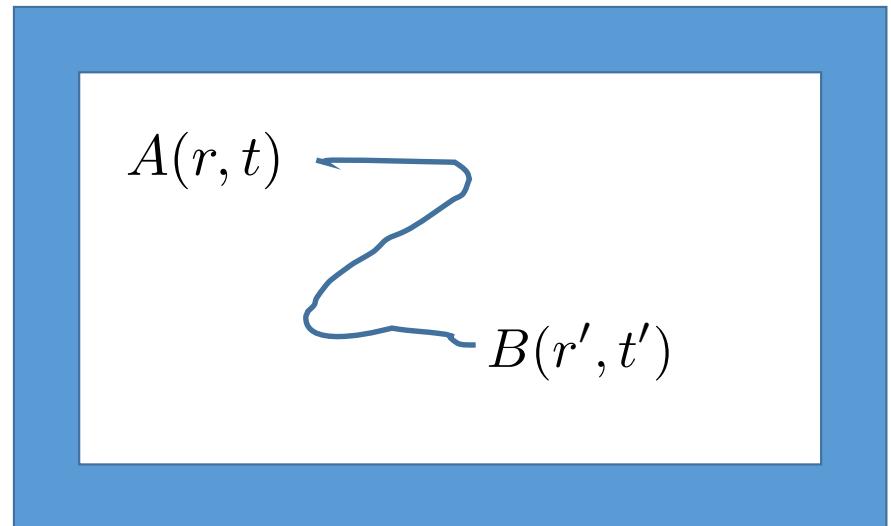
$$\langle j(r, t)j(r', t') \rangle$$

Single-particle spectra (ARPES)

$$\langle c(r, t)c^\dagger(r', t') \rangle$$

Spin-resolved neutron scattering

$$\sigma_{\alpha\beta}^{x,y,z} \langle S_\alpha(r, t)S_\beta(r', t') \rangle$$

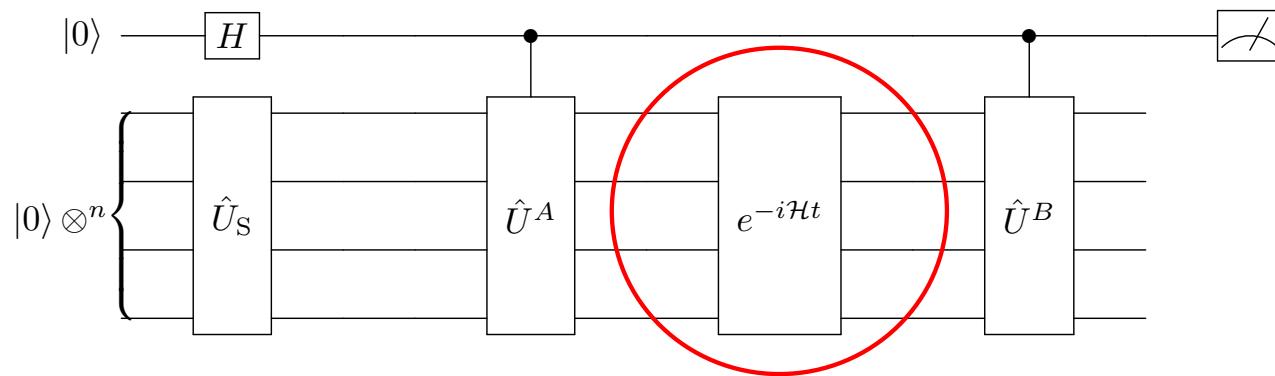


# Low-energy excitations: correlation functions

Express the correlation function through the Lehmann representation:

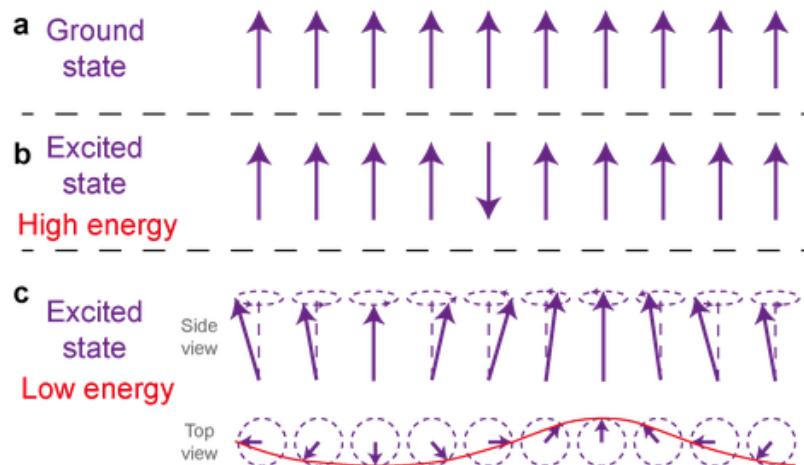
$$C(t) = \langle \Phi | \hat{U}^B(t) \hat{U}^A(0) | \Phi \rangle = \sum_m e^{-i(E_m - E_0)t} \langle \phi_0 | U^B | m \rangle \langle m | U^A | \phi_0 \rangle.$$

Quantum circuit:

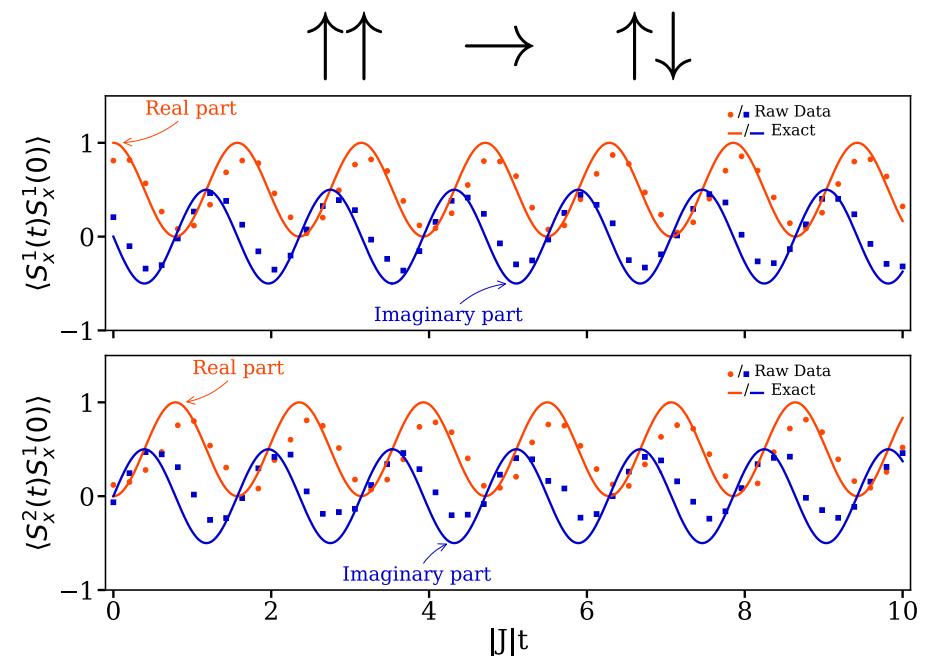


# Low-energy excitations: 2-site magnons

Spin-spin correlation function for periodic Heisenberg model: Magnons!



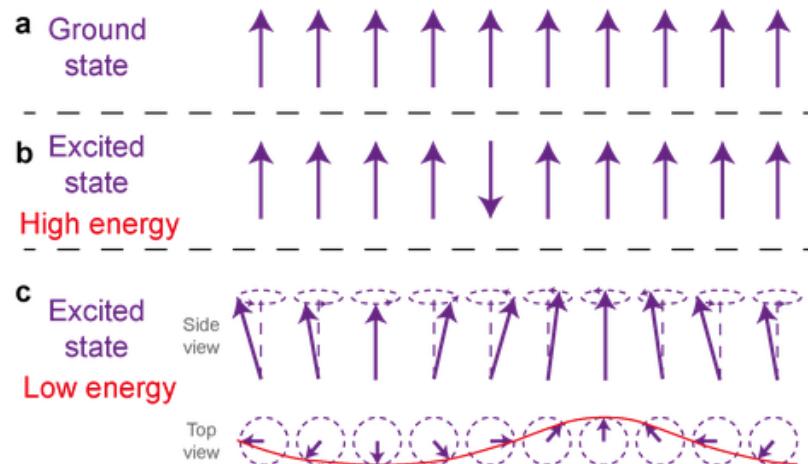
$$\hat{H} = 2SJ \sum_k (1 - \cos(k)) \hat{c}_k^\dagger \hat{c}_k$$



Data from *ibmq\_tokyo*

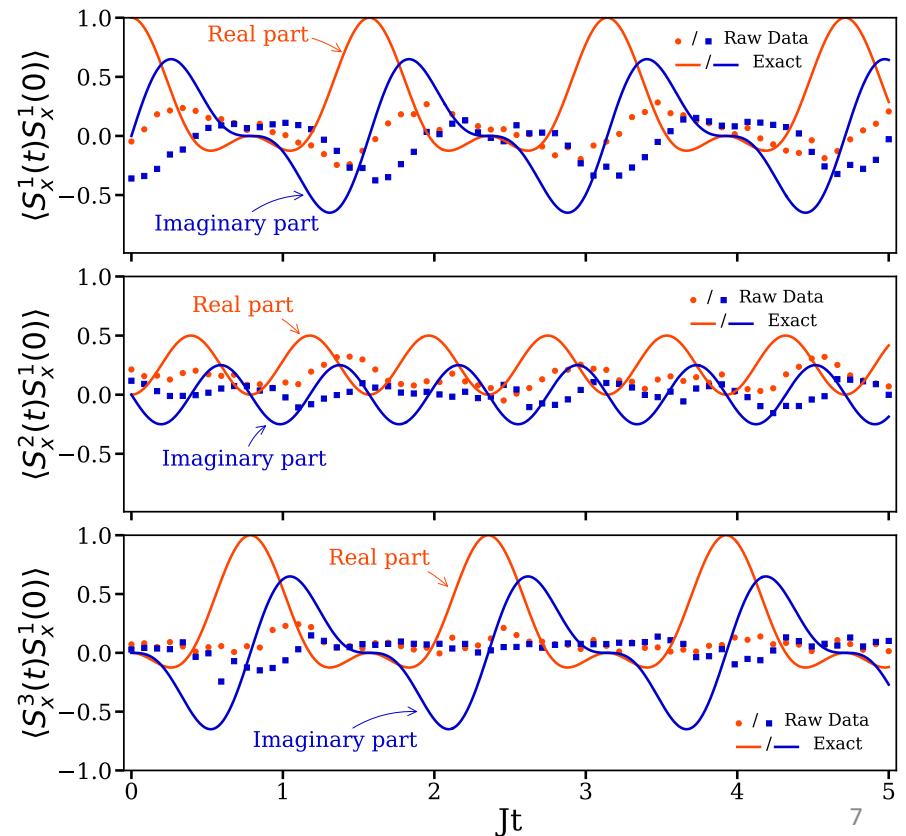
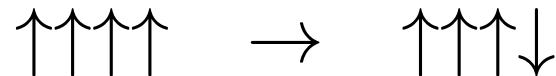
# Low-energy excitations: 4-site magnons

Spin-spin correlation function for periodic Heisenberg model

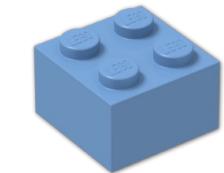
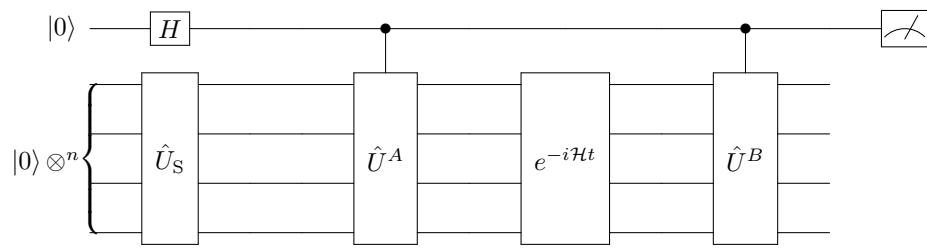


$$\hat{H} = 2SJ \sum_k (1 - \cos(k)) \hat{c}_k^\dagger \hat{c}_k$$

Data from *ibmq\_tokyo*



# Quantum Compiling



Single Qubit Gates

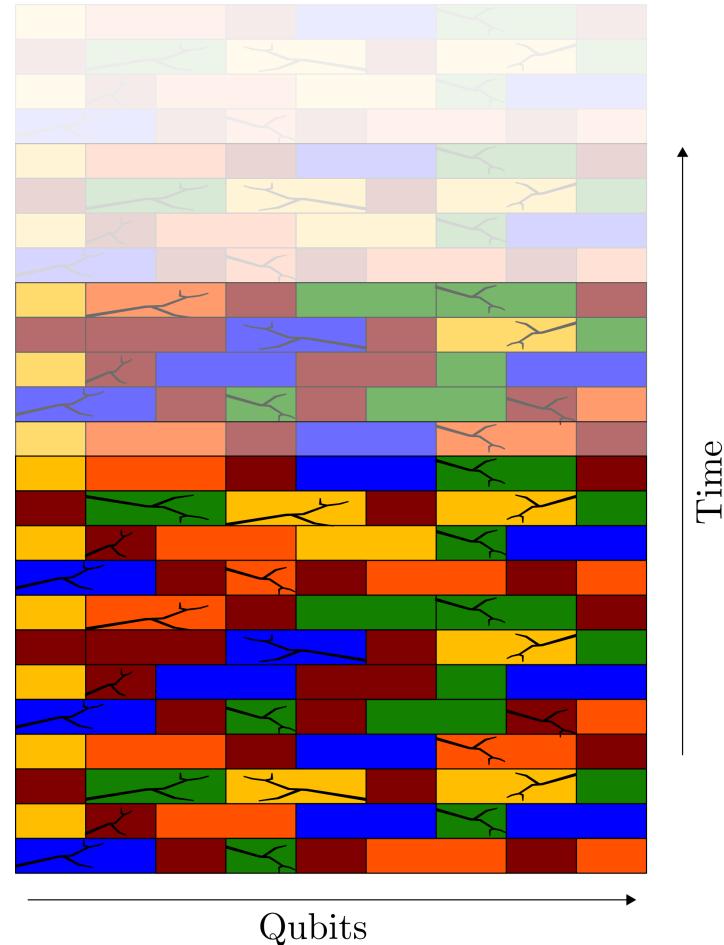
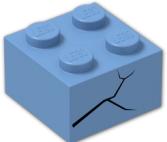
+



Two Qubit Gate

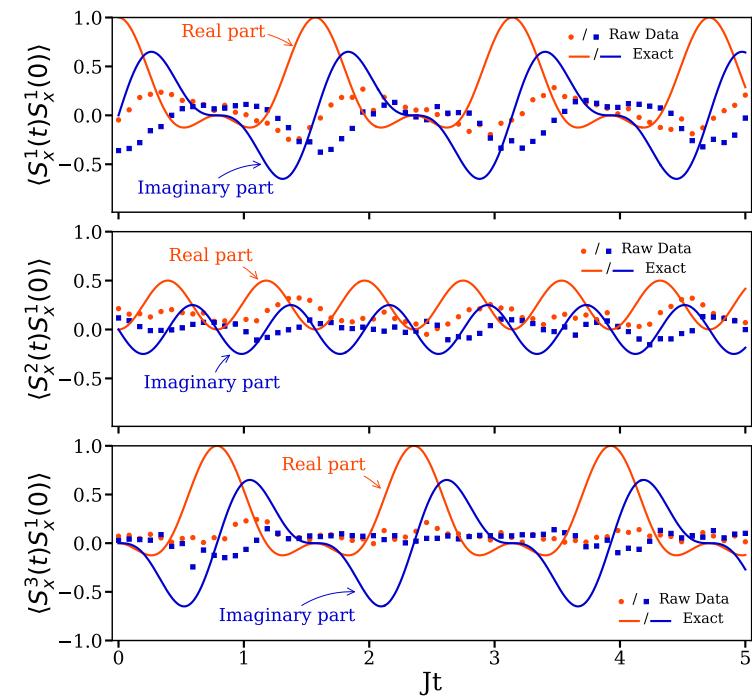
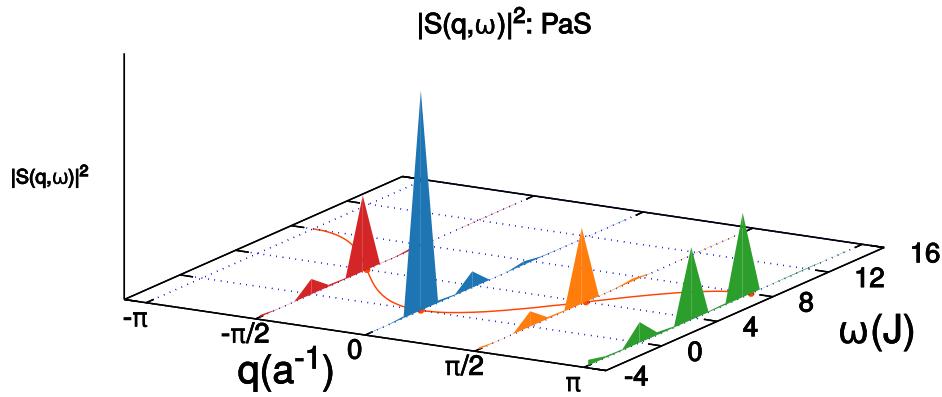
= Anything!\*

\*Actual Gates



# Low-energy excitations: 4-site magnons

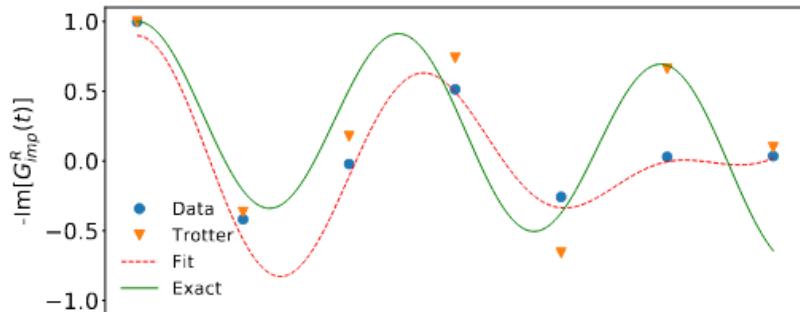
Spin-spin correlation function for periodic Heisenberg model: Magnons!



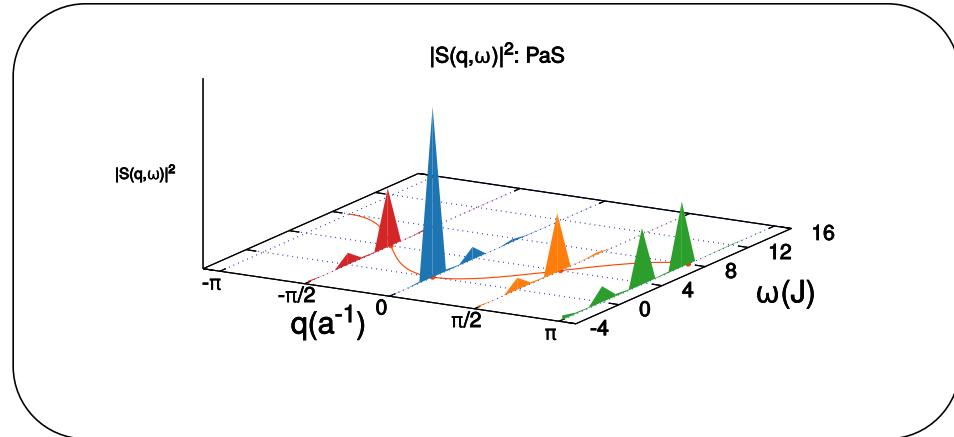
[10.1103/PhysRevB.101.014411](https://doi.org/10.1103/PhysRevB.101.014411)

# Low-energy excitations: 4-site magnons

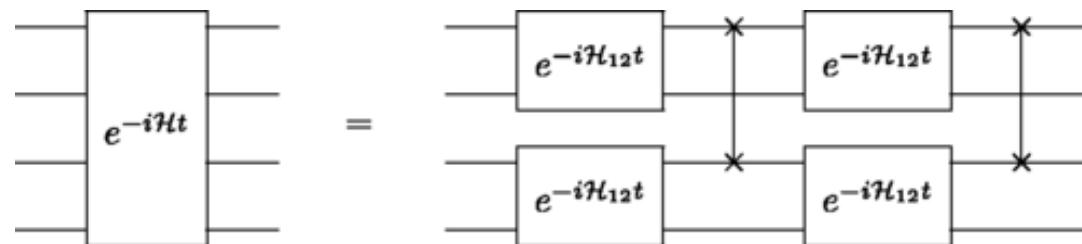
Q: Why does this work?



Keen et al., QST [10.1088/2058-9565/ab7d4c](https://doi.org/10.1088/2058-9565/ab7d4c)



A: Constant depth time evolution circuits

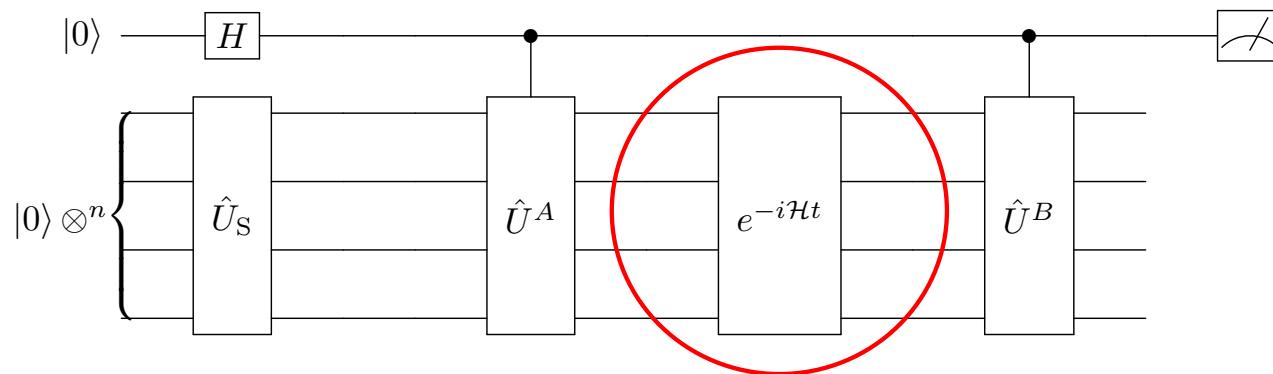


# Low-energy excitations: correlation functions

Express the correlation function through the Lehmann representation:

$$C(t) = \langle \Phi | \hat{U}^B(t) \hat{U}^A(0) | \Phi \rangle = \sum_m e^{-i(E_m - E_0)t} \langle \phi_0 | U^B | m \rangle \langle m | U^A | \phi_0 \rangle.$$

Quantum circuit:



## Trotter Approximation

Consider 5 spin Kitaev chain:



$$\mathcal{H} = a XXIII + b IYYII + c IIXXI + d IIIYY$$

$$U(t) = e^{-it\mathcal{H}} \neq e^{-ita} XXIII e^{-itb} IYYII e^{-itc} IIXXI e^{-itd} IIIYY$$

## Trotter Approximation

Consider 5 spin Kitaev chain:



$$\mathcal{H} = a \text{ } XXIII + b \text{ } IYYII + c \text{ } IIIXXI + d \text{ } IIIYY$$

$$U(\epsilon) = e^{-i\epsilon\mathcal{H}} = e^{-i\epsilon a} XXIII e^{-i\epsilon b} IYYII e^{-i\epsilon c} IIIXXI e^{-i\epsilon d} IIIYY + O(\epsilon^2)$$

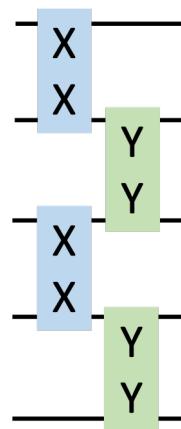
## Trotter Approximation

Consider 5 spin Kitaev chain:



$$\mathcal{H} = a XXIII + b IYYII + c IIXXI + d IIIYY$$

$$U(\epsilon) = e^{-i\epsilon\mathcal{H}} = e^{-i\epsilon a} XXIII e^{-i\epsilon b} IYYII e^{-i\epsilon c} IIXXI e^{-i\epsilon d} IIIYY + O(\epsilon^2)$$



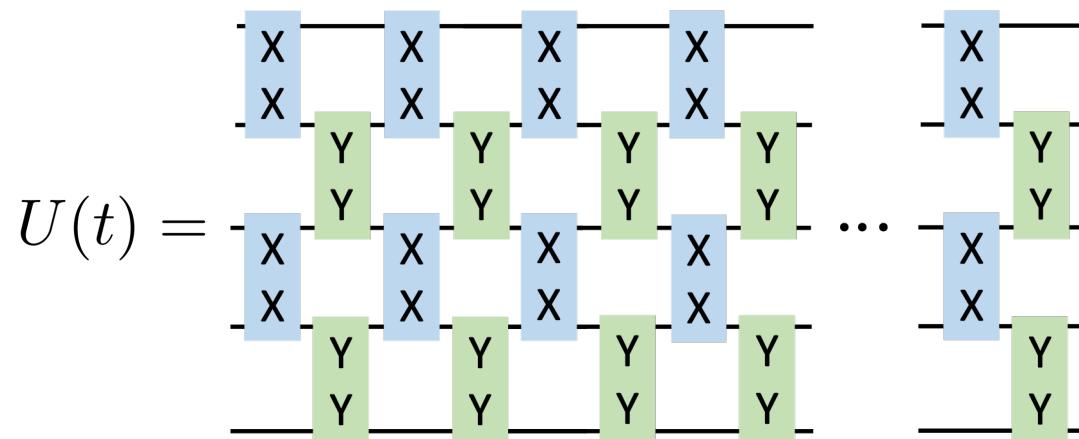
## Trotter Approximation

Consider 5 spin Kitaev chain:



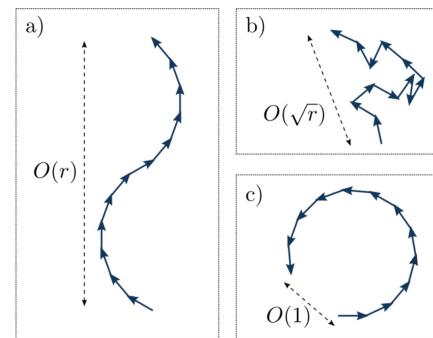
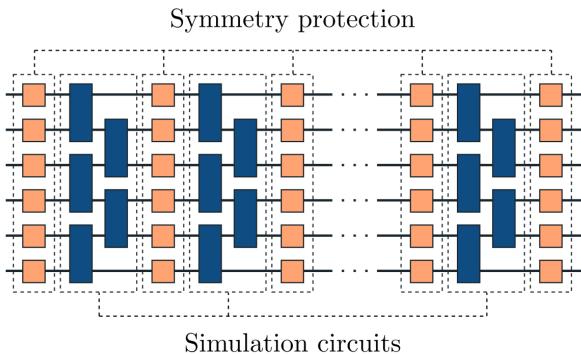
$$\mathcal{H} = a XXIII + b IYYII + c IIXXI + d IIIYY$$

$$U(\epsilon) = e^{-i\epsilon\mathcal{H}} = e^{-i\epsilon a} XXIII e^{-i\epsilon b} IYYII e^{-i\epsilon c} IIXXI e^{-i\epsilon d} IIIYY + O(\epsilon^2)$$



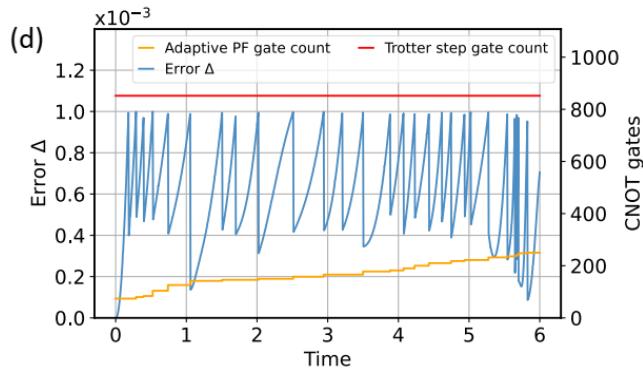
# Improving Trotter Expansions

- Symmetry protection to reduce trotter error (\*)



- Adaptive product formula to reduce number of trotter steps needed (\*\*)

$$e^{-iH\delta t}|\Psi(t)\rangle \approx T(\vec{O}', \vec{\Lambda}', t)|\Psi_0\rangle$$

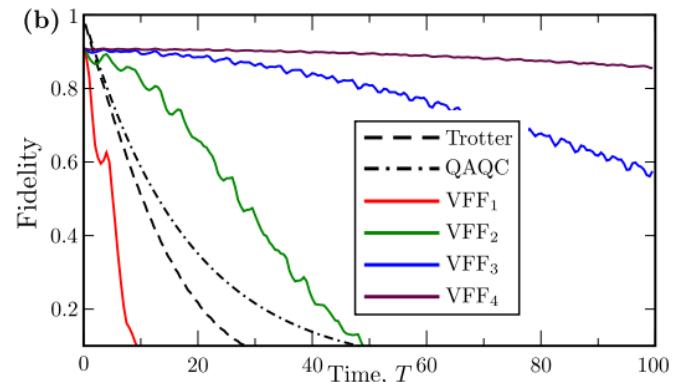
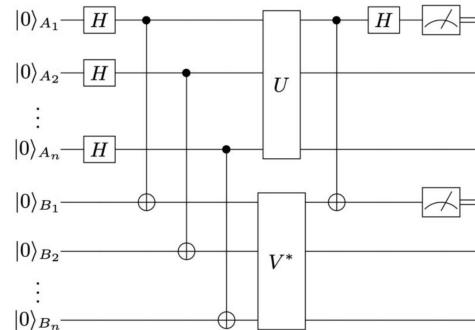
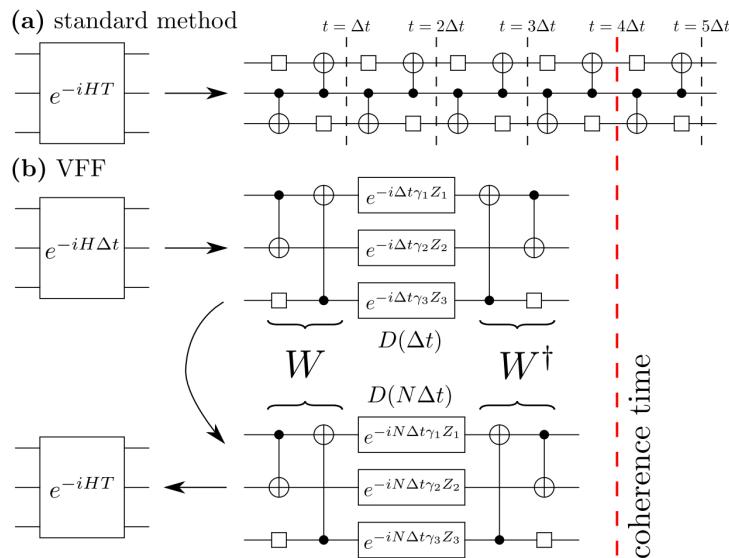


(\*) Tran, M. C., Su, Y., Carney, D., & Taylor, J. M. (2021), PRX Quantum, 2(1), 010323.

(\*\*) Zhang, Z. J., Sun, J., Yuan, X., & Yung, M. H. (2020), arXiv:2011.05283.

# Variational Fast Forwarding

- QAQC on one trotter step to diagonalize the Hamiltonian
- then simulate for whichever simulation time you want with a fixed depth circuit!



(\*) Cirstoiu, C., Holmes, Z., Iosue, J., Cincio, L., Coles, P. J., & Sornborger, A. (2020),  
npj Quantum Information, 6(1), 1-10.

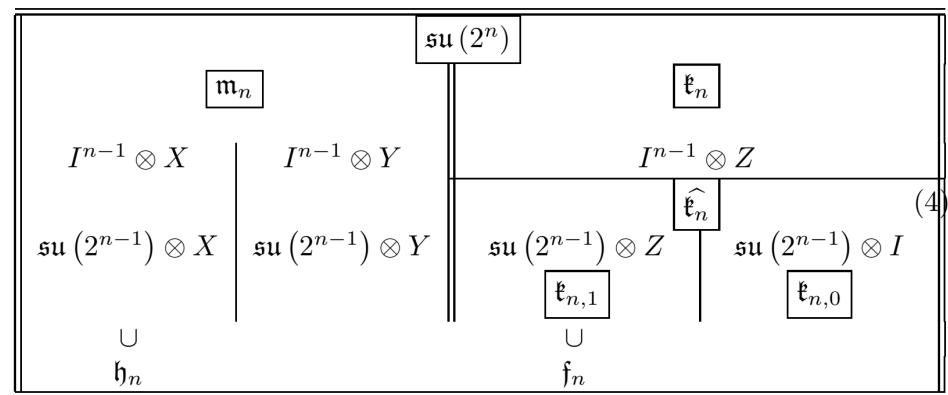
# Unitary Synthesis: Cartan Decomposition

- Cartan decomposition found its application in generic unitary synthesis for quantum circuits  $(*, **)$

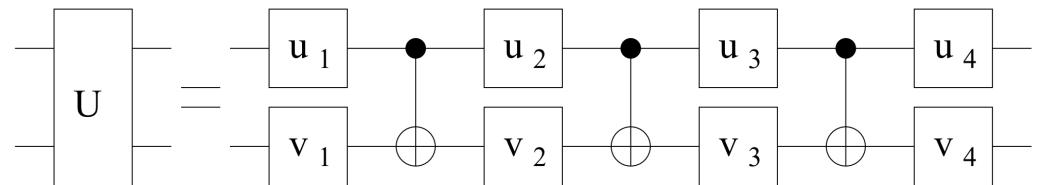
$$\mathfrak{g} = \mathfrak{m} \oplus \mathfrak{k}$$

$$\begin{aligned} [\mathfrak{k}, \mathfrak{k}] &\subset \mathfrak{k} \\ [\mathfrak{m}, \mathfrak{k}] &= \mathfrak{m} \\ [\mathfrak{m}, \mathfrak{m}] &\subset \mathfrak{k}. \end{aligned}$$

- It is optimal for SU(4) (2 qubits)!  $(***)$



$$I^{n-1} = I^{\otimes(n-1)} = \underbrace{I \otimes \dots \otimes I}_{n-1}$$



(\*) N. Khaneja and S. J. Glaser, Chemical Physics 267, 11 (2001).

(\*\*) H. N. Sa Earp and J. K. Pachos, Journal of Mathematical Physics 46, 082108 (2005), doi.org/10.1063/1.2008210.

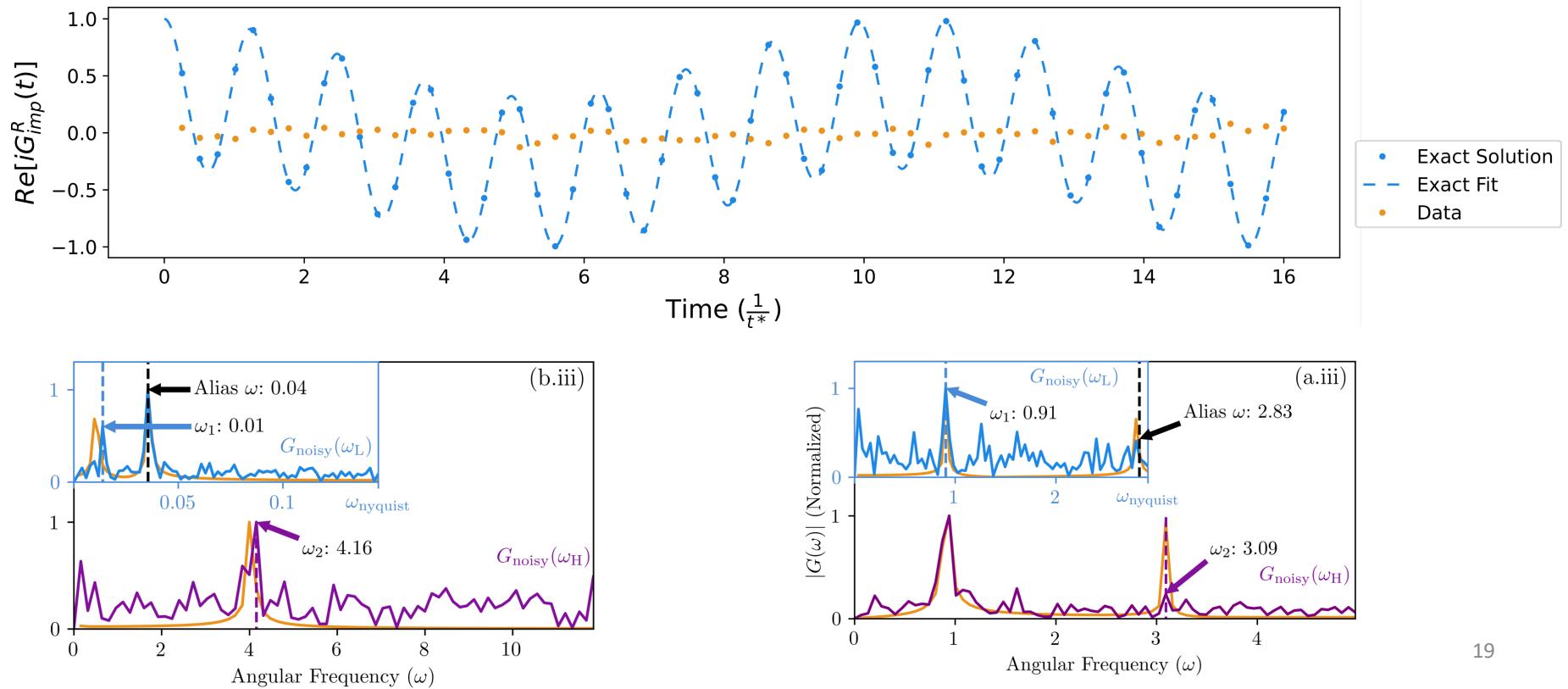
(\*\*\*) G. Vidal and C. M. Dawson, Physical Review A 69, 010301 (2004).

# What can you do with fixed depth time evolution circuits?

[T. Steckmann et al., arXiv:2112.05688](https://arxiv.org/abs/2112.05688)

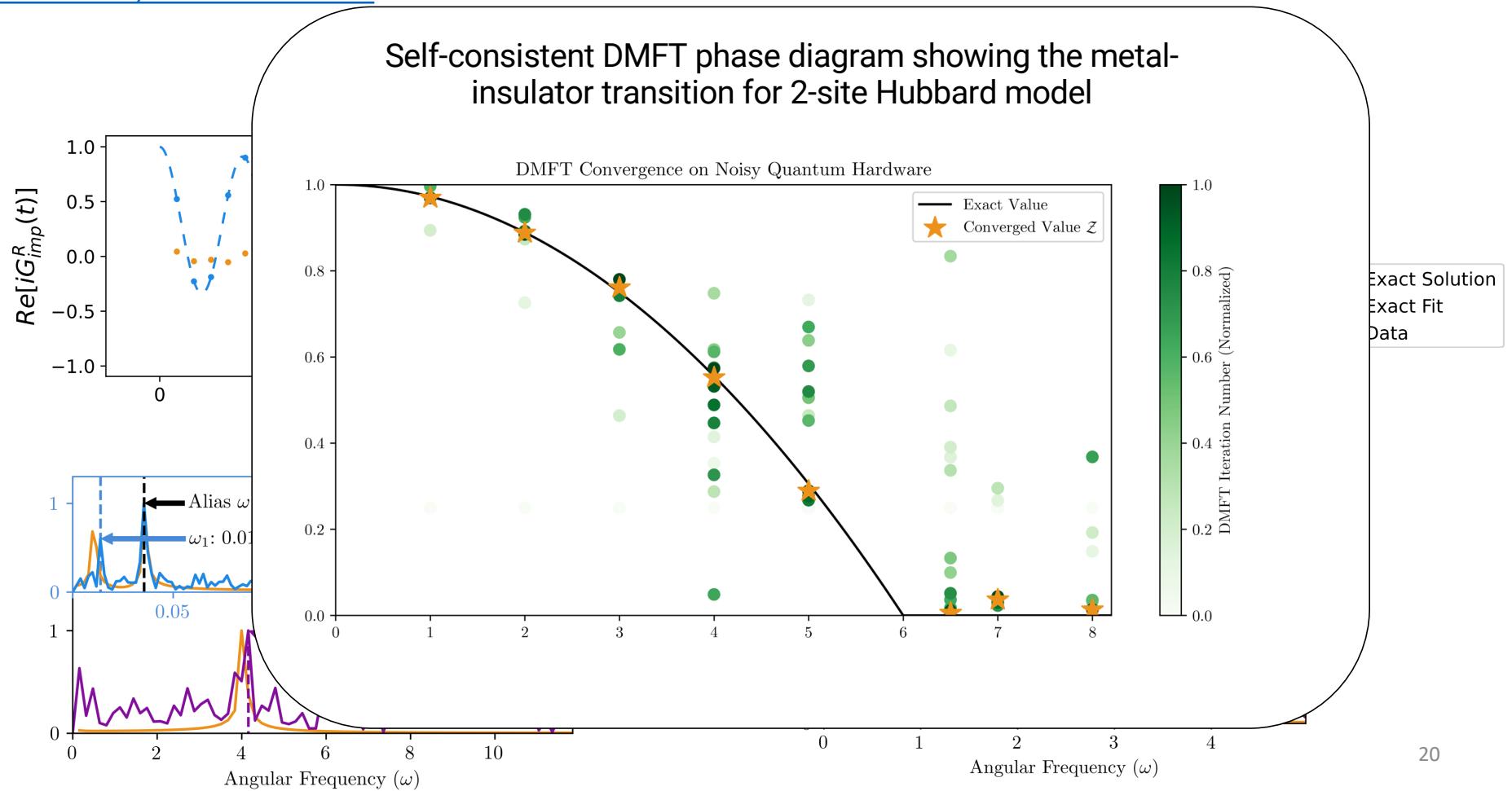
## 2-site Hubbard DMFT (5 qubits)

Cartan Based Simulation on IBM Lagos



## 2-site Hubbard DMFT

T. Steckmann et al., arXiv:2112.05688



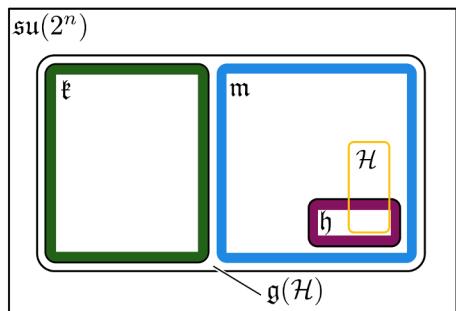
## 2 Algebraic methods for circuit compression

Cartan Decomposition

Algebraic Compression

## 2 Algebraic methods for circuit compression

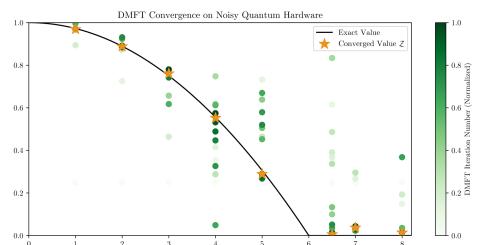
### Cartan Decomposition



- Produces exact, fixed depth time evolution unitaries for any model.
- Produces unitaries for linear combinations of (anti)-Hermitian operators (UCC factors).
- We have code available!  
<https://github.com/kemperlab/cartan-quantum-synthesizer>

Phys. Rev. Lett. 129, 070501 (2022) , arXiv:2112.05688

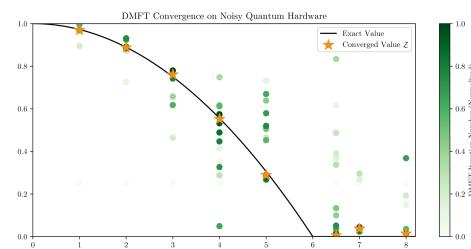
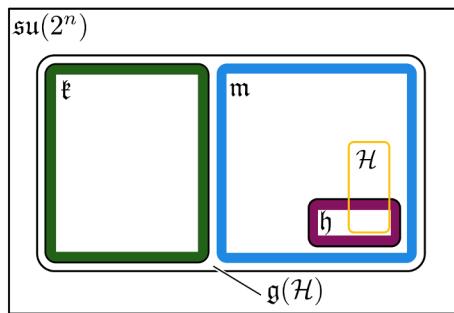
### Algebraic Compression



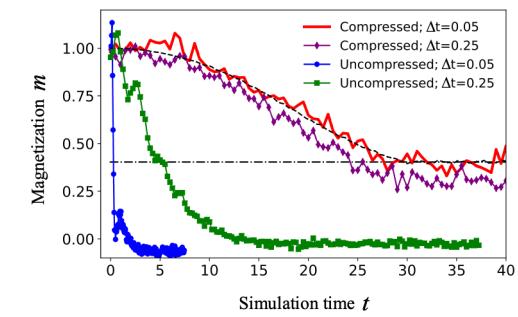
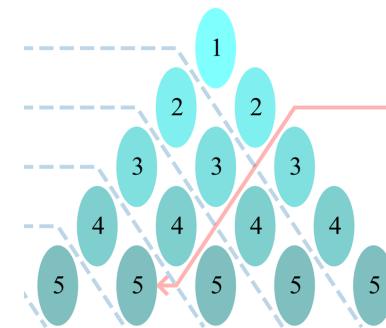
Phys. Rev. A 105, 032420 (2021), SIMAX 2022 43:3, 1084-1108

## 2 Algebraic methods for circuit compression

### Cartan Decomposition



### Algebraic Compression



- Produces exact, fixed depth time evolution unitaries for any model.
- Produces unitaries for linear combinations of (anti)-Hermitian operators (UCC factors).
- We have code available!  
<https://github.com/kemperlab/cartan-quantum-synthesizer>

- Compressed Trotter circuits down to a shallow fixed depth circuit for 1-D nearest neighbor TFXY, TFIM, XY and Kitaev models.
- Based on 3 easy to check, local properties.
- We have code available! Check F3C, F3C++ and F3Cpy at <https://github.com/QuantumComputingLab>

# Approach #1: Cartan Decomposition

## Main Problem

Exact simulation of a time independent spin Hamiltonian:  $\mathcal{H} = \sum_j h_j \sigma^j$

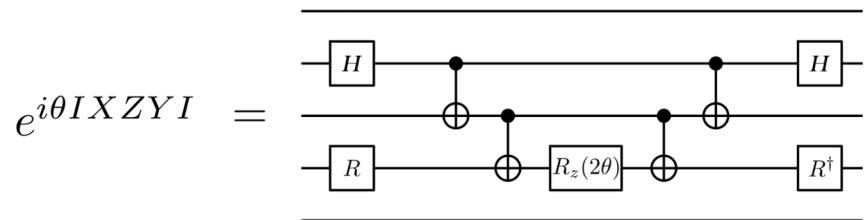
Time evolution operator:

Single exponential circuit:

Two main issues: 1)  $4^n - 1$  many  $\kappa_i$

2) What cost function? Norm of the difference?

$$U(t) = e^{-it\mathcal{H}} = \prod_{\bar{\sigma}^i \in \mathfrak{su}(2^n)} e^{i\kappa_i \bar{\sigma}^i}$$



# Hamiltonian Algebra

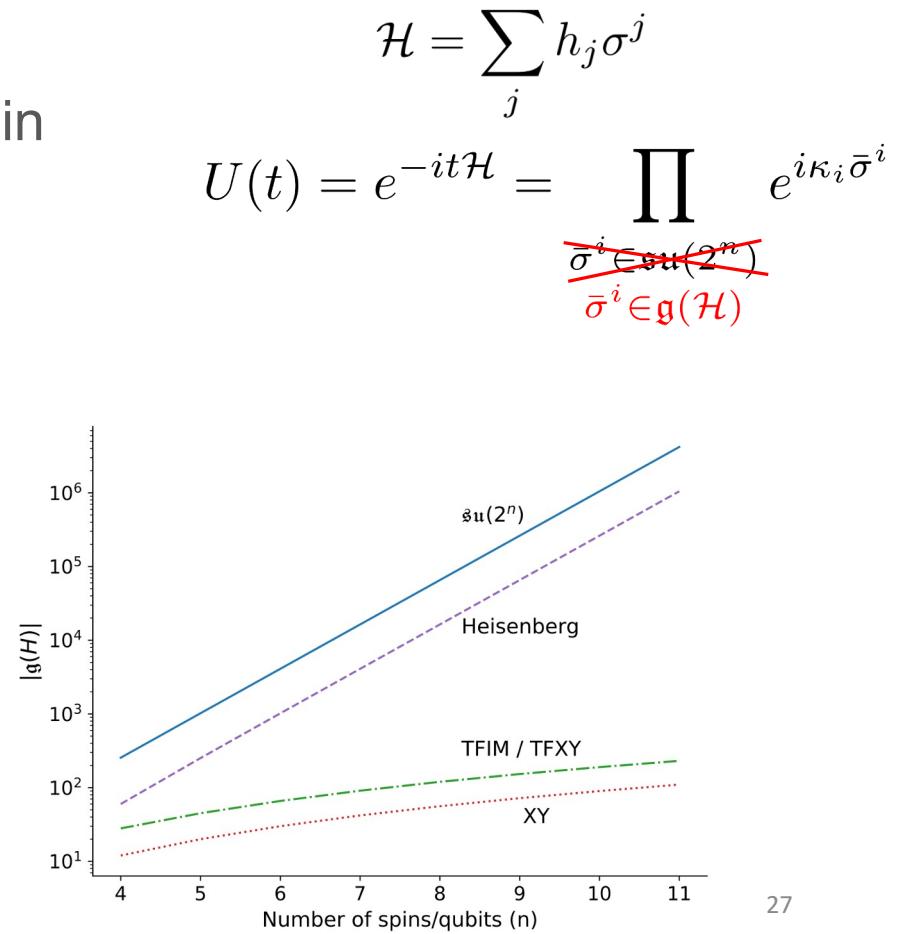
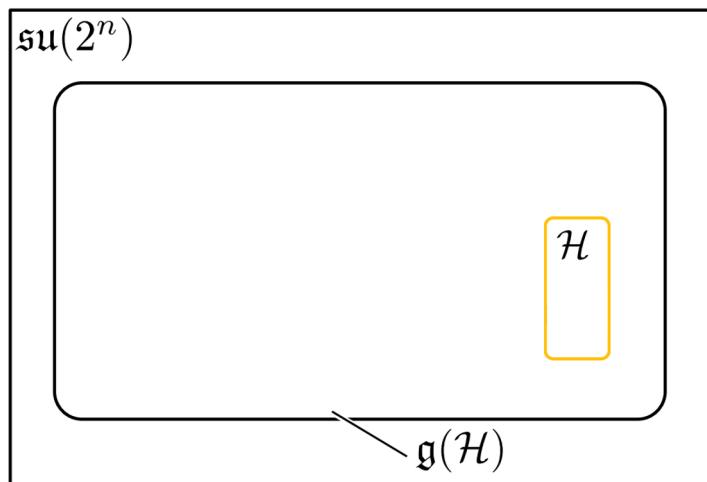
- We don't have to work in full  $\mathfrak{su}(2^n)$

$$\mathcal{H} = \sum_j h_j \sigma^j$$

$$U(t) = e^{-it\mathcal{H}} = \prod_{\bar{\sigma}^i \in \mathfrak{su}(2^n)} e^{i\kappa_i \bar{\sigma}^i}$$

# Hamiltonian Algebra

- We don't have to work in full  $\mathfrak{su}(2^n)$
- Get the closure of the Pauli strings within the Hamiltonian under commutation i.e. the “Hamiltonian algebra”  $\mathfrak{g}(\mathcal{H})$



## Main Problem

Exact simulation of a time independent spin Hamiltonian:  $\mathcal{H} = \sum_j h_j \sigma^j$

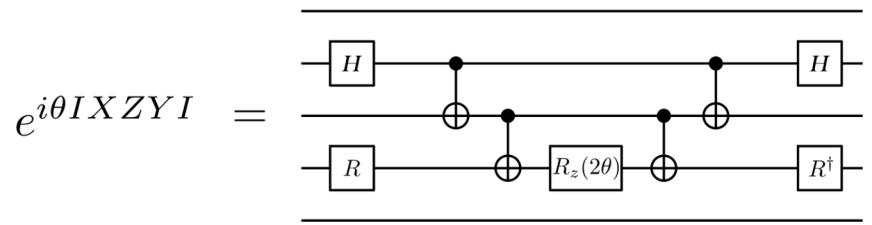
Time evolution operator is:

Single exponential circuit is given as:

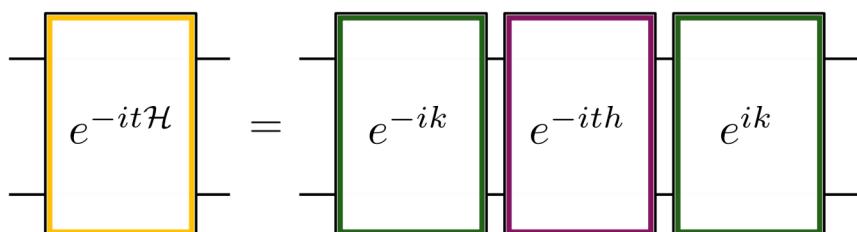
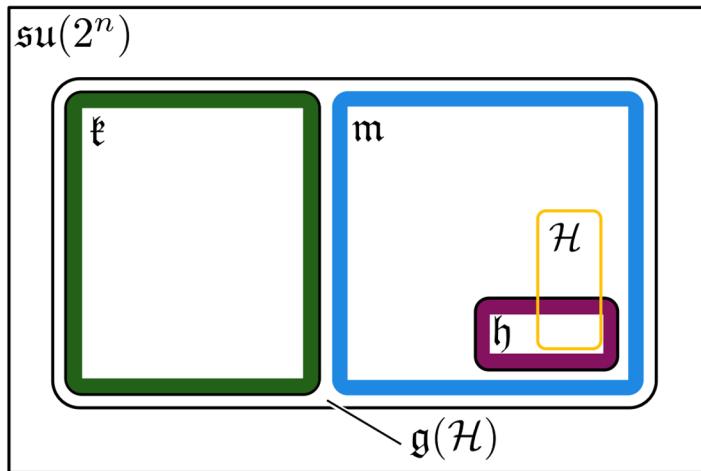
Two main issues: 1)  $4^n - 1$  many  $\kappa_i$

2) What cost function? Norm of the difference?

$$U(t) = e^{-it\mathcal{H}} = \prod_{\substack{\bar{\sigma}^i \in \mathfrak{su}(2^n) \\ \bar{\sigma}^i \in \mathfrak{g}(\mathcal{H})}} e^{i\kappa_i \bar{\sigma}^i}$$



# Cartan Decomposition and KHK Theorem



Have  $H \in \mathfrak{m}$ , and consider the following function

$$f(K) = \langle KvK^\dagger H \rangle$$

where

$$K = e^{\theta_1 k_1} e^{\theta_2 k_2} \dots e^{\theta_{n_h} k_{n_h}}$$

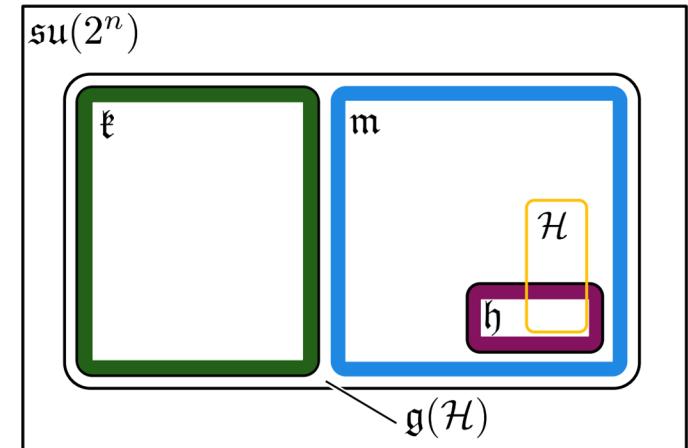
$$v = h_1 + \pi h_2 + \pi^2 h_3 + \dots + \pi^{n_h-1} h_{n_h}$$

Then for any local minimum or maximum of the function  $f$  denoted by  $K_0$  will satisfy

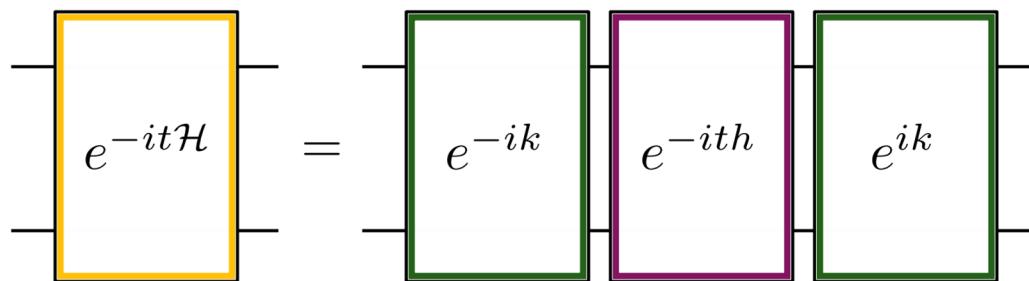
$$K_0^\dagger H K_0 \in \mathfrak{h}$$

# Algorithm

- 1) Generate Hamiltonian algebra  $g(\mathcal{H})$
- 2) Find a Cartan decomposition where  $H$  is in  $m$
- 3) Obtain parameters via local minimum of  $f(K)$
- 4) Build the circuit using  $K$  and  $h$
- 5) Then simulate for any time you want!

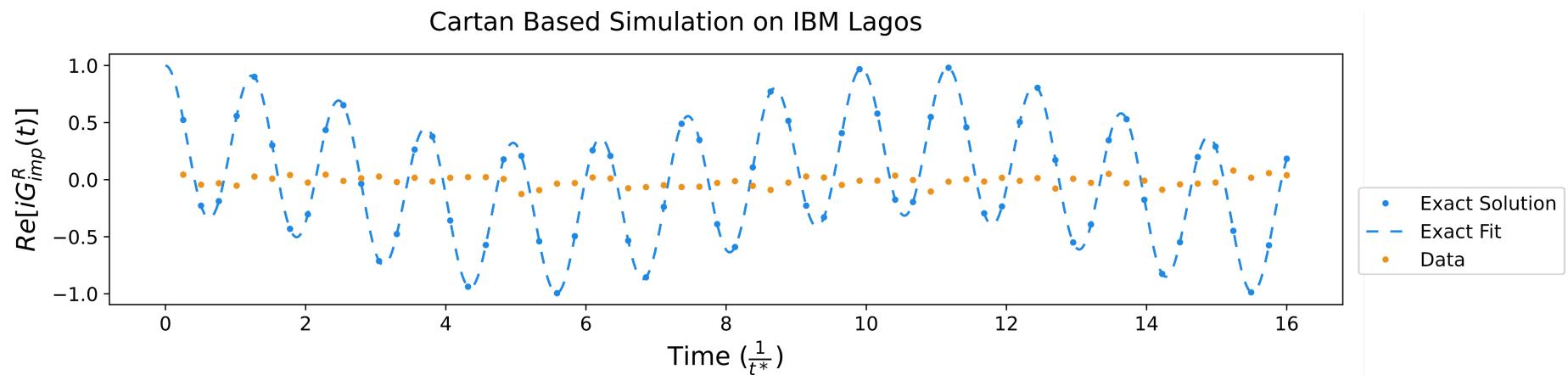
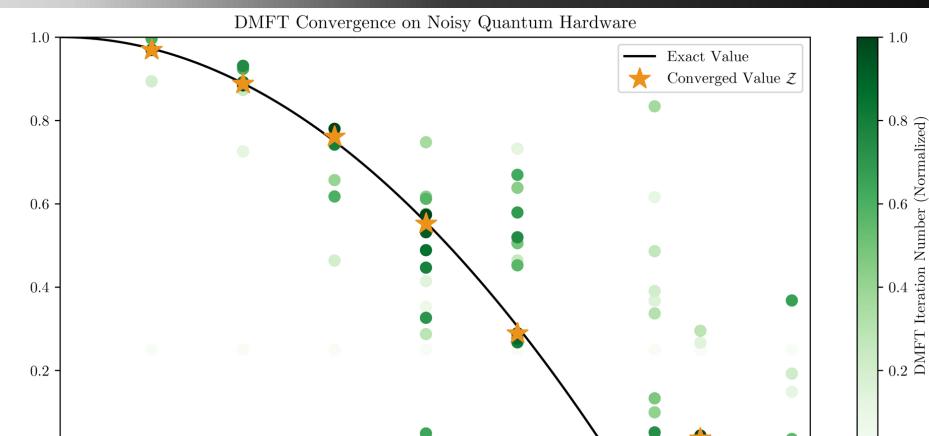


$$f(K) = \langle KvK^\dagger, \mathcal{H} \rangle$$



## Approach #1: Cartan Decomposition

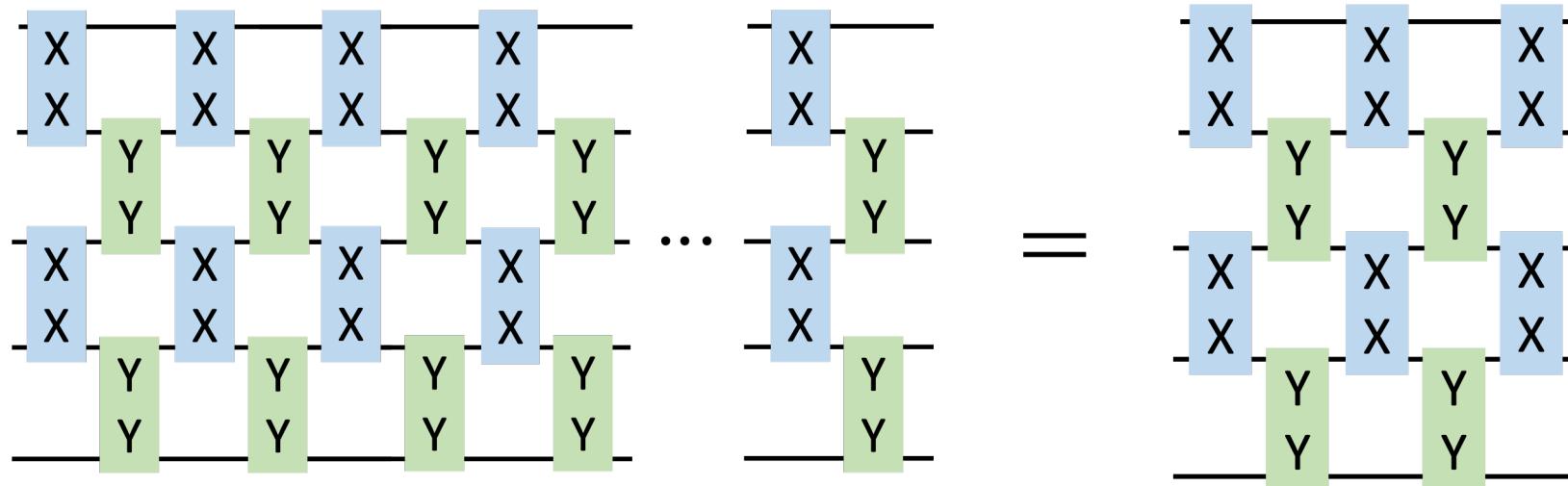
- $O(n^2)$  circuit for TFIM, TFXY, XY
- Applicable for any model
- Optimize only once for any time t
- Obtained 1<sup>st</sup> ever self-consistent DMFT Hubbard phase diagram on IBM QC.



# Approach #2: Algebraic Circuit Compression

## Approach #2: Algebraic Circuit Compression

- We propose a **constructive**, Lie algebra based method which leads to fixed depth circuits for several models
- The method is **scalable** due to its “constructive” and “local” nature.



## Approach #2: Algebraic Circuit Compression

We define an abstract object called “block” which satisfies:

**Fusion**

$$\begin{array}{c} i \\ \text{---} \\ i \end{array} = \begin{array}{c} i \end{array}$$

**Commutation**

$$\begin{array}{c} i \\ \text{---} \\ i+2 \end{array} = \begin{array}{c} i \\ \text{---} \\ i+2 \end{array}$$

**Turnover**

$$\begin{array}{c} i \\ \text{---} \\ i+1 \end{array} = \begin{array}{c} i \\ \text{---} \\ i+1 \end{array} \begin{array}{c} i \\ \text{---} \\ i \end{array}$$

Blocks will be mapped to certain quantum gates in a model specific way.

These properties are **local!** One needs to check only the neighbor gates to apply them, not the whole circuit.

## Approach #2: Algebraic Circuit Compression

### Fusion

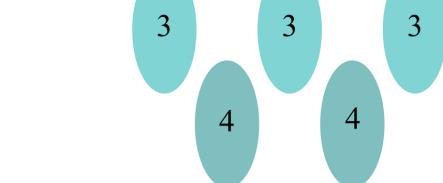
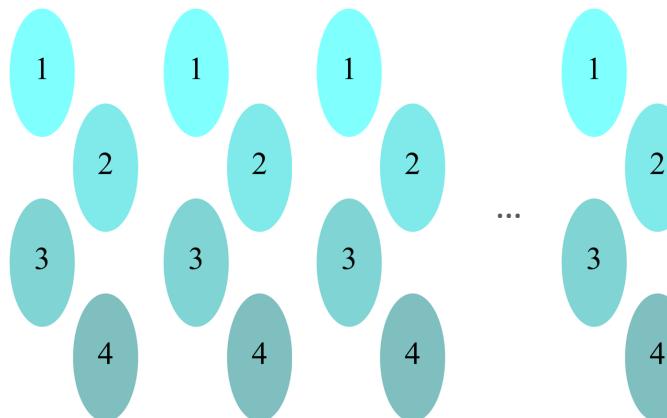
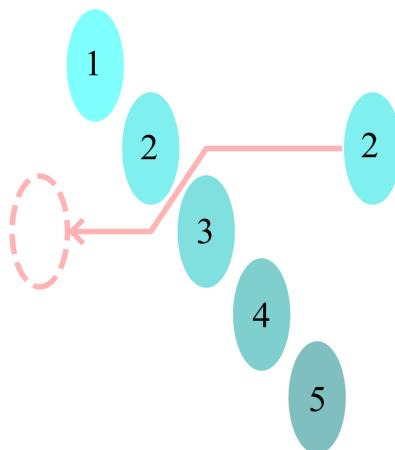
$$\begin{array}{c} i \\ \text{---} \\ i \end{array} = \begin{array}{c} i \end{array}$$

### Commutation

$$\begin{array}{c} i \\ \text{---} \\ i+2 \end{array} = \begin{array}{c} i \\ \text{---} \\ i+2 \end{array}$$

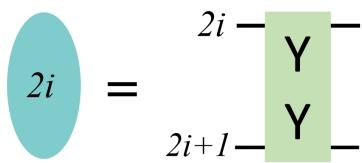
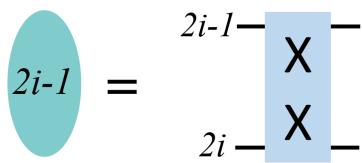
### Turnover

$$\begin{array}{c} i \\ \text{---} \\ i+1 \end{array} = \begin{array}{c} i \\ \text{---} \\ i+1 \end{array} \quad \begin{array}{c} i \\ \text{---} \\ i+1 \end{array}$$



## Examples

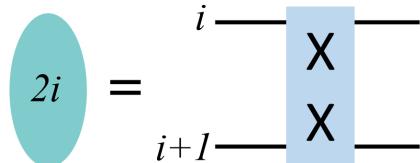
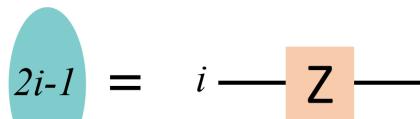
Kitaev Chain



$n(n-1)/2$  XX gates

$n(n-1)$  CNOTs

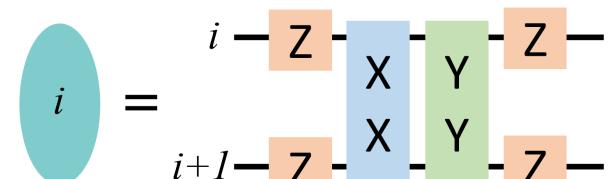
Transverse Field Ising



$n(n-1)$  XX gates

$2n(n-1)$  CNOTs

Transverse Field XY

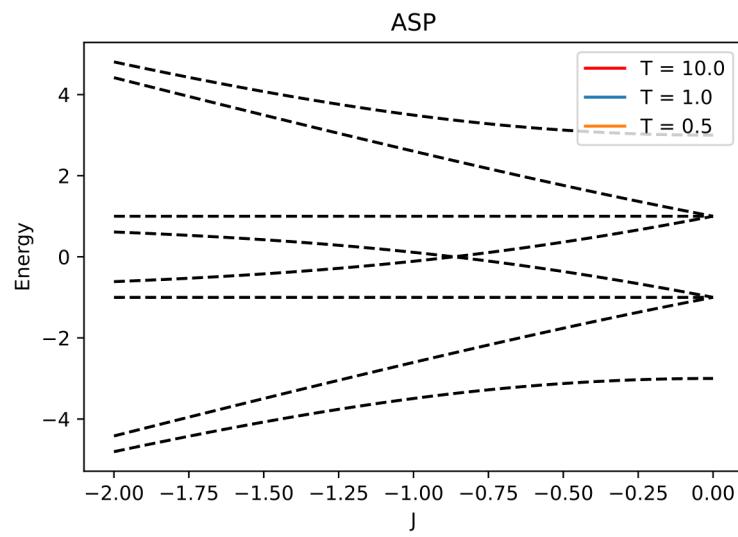


$n(n-1)$  XX gates

$n(n-1)$  CNOTs

## Main Example: Ising Adiabatic State Preparation

$$H = -2(X_1X_2 + X_2X_3) - (Z_1 + Z_2 + Z_3)$$



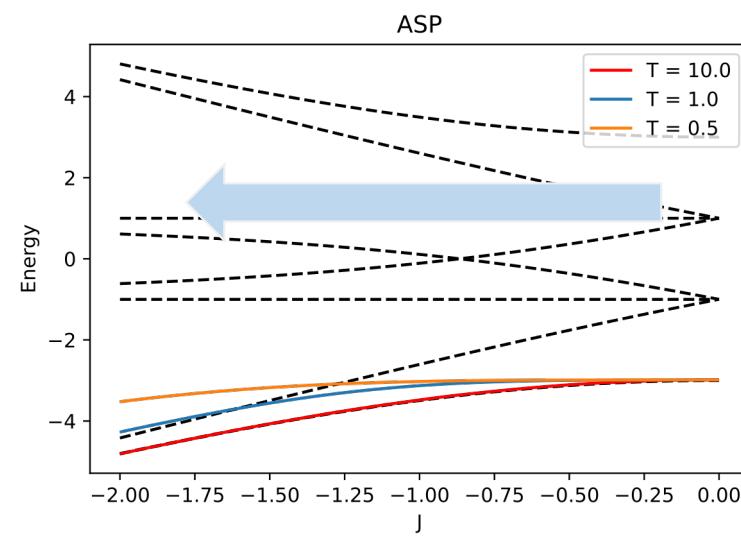
$$H = -(Z_1 + Z_2 + Z_3)$$

$$|\psi\rangle = |000\rangle$$

$$H = J(t)(X_1X_2 + X_2X_3) - (Z_1 + Z_2 + Z_3)$$

## Main Example: Adiabatic State Preparation

$$H = -2(X_1X_2 + X_2X_3) - (Z_1 + Z_2 + Z_3)$$



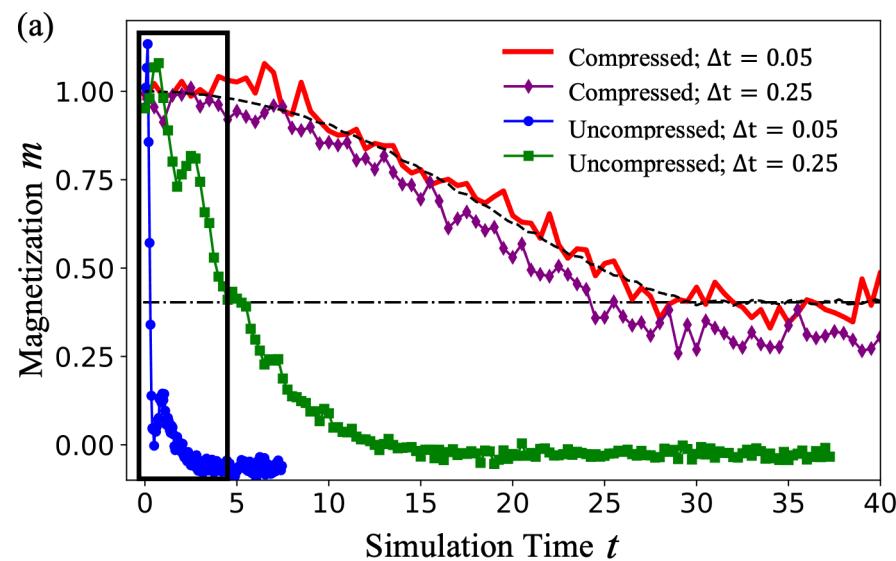
$$H = -(Z_1 + Z_2 + Z_3)$$

$$|\psi\rangle = |000\rangle$$

$$H = J(t)(X_1X_2 + X_2X_3) - (Z_1 + Z_2 + Z_3)$$

## Approach #1: Algebraic Circuit Compression

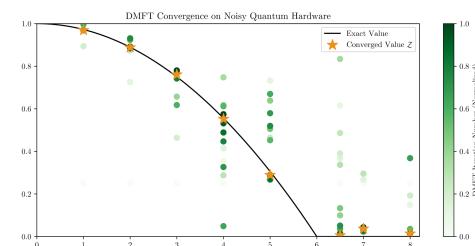
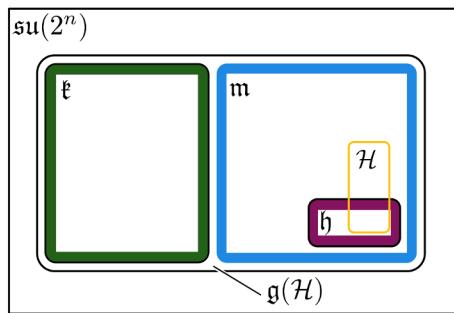
$$\mathcal{H}_{ASP}(t) = J(t) \sum_{i=1}^{n-1} X_i X_{i+1} + h_z \sum_{i=1}^n Z_i \quad \langle m(t) \rangle \equiv \frac{1}{n} \sum_i \sigma_i^z(t)$$



- Compressed circuits have 20 CNOT gates in total whereas Trotter circuits have increasing number of CNOTs as simulation time increases

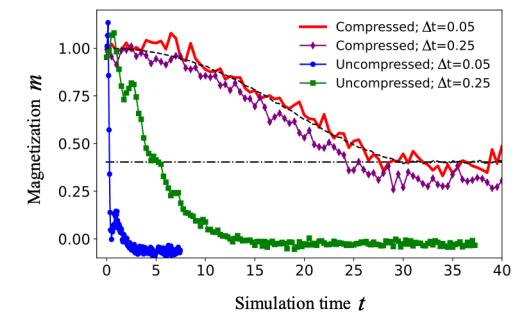
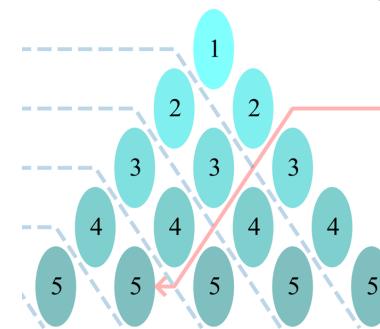
## 2 Algebraic methods for circuit compression

### Cartan Decomposition



- Produces exact, fixed depth time evolution unitaries for any model.
- Produces unitaries for combinations of (anti)-Hermitian operators (UCC factors).
- We have code available!  
<https://github.com/kemperlab/cartan-quantum-synthesizer>

### Algebraic Compression



- Compressed Trotter circuits down to a fixed depth circuit for 1-D nearest neighbor TFXY, TFIM, XY and Kitaev models.
- Based on 3 easy to check, local properties.
- We have code available! Check F3C, F3C++ and F3Cpy at <https://github.com/QuantumComputingLab>