Event generation with quantum computers through particle-oriented simulation

Yutaro Iiyama
ICEPP, The University of Tokyo

CHEP 2024
October 23, 2024, Krakow

Let’s develop a quantum event generator!

Why quantum?

Fundamental scaling problems in generators:

- Event complexity scales ~factorially with perturbation order
- Integration time scales ~exponentially with final-state multiplicity

Timing and memory usage (Sherpa 3.x.y + HDF5)

LO ME level event generation only (Comix; $\gamma, Z, h, \mu, \nu_\mu, \tau, \nu_\tau$ off)

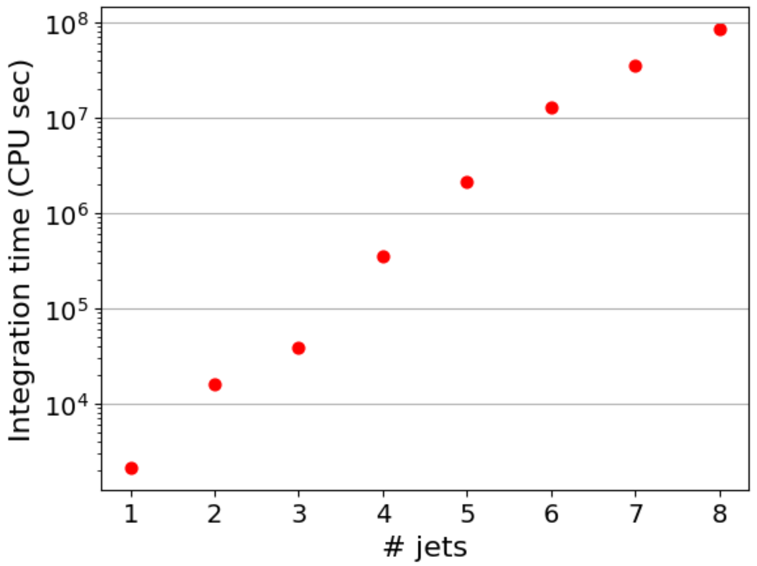
Process $W^+ +$	1j	2j	3j	4j
RAM Usage	21 MB	43 MB	48 MB	85 MB
Init/startup time	<1s / <1s	<1s / <1s	2s / <1s	32s / <1s
Integration time	8×4m26s	16×16m42s	32×20m26s	64×1h32m
MC uncertainty	0.22%	0.46%	0.89%	0.97%
Unweighting eff	$6.59 \cdot 10^{-3}$	$7.50 \cdot 10^{-4}$	$2.71 \cdot 10^{-4}$	$1.47 \cdot 10^{-4}$
10k evts	1m 2s	15m 5s	1h 3m	5h 56m

Numbers generated on dual 8-core Intel® Xeon® E5-2660 @ 2.20GHz

Process $W^+ +$	5j	6j*	7j*	8j†
RAM Usage	189 MB	484 MB	1.32 GB	1.32 GB
Init/startup time	3m5s / 1s	24m52s / 5s	3h6m / 16s	5h55m / 29s
Integration time	128×4h38m	256×13h53m	512×19h0m	1024×23h8m
MC uncertainty	1.0%	0.99%	2.38%	4.68%
Unweighting eff	$9.56 \cdot 10^{-5}$	$7.66 \cdot 10^{-5}$	$7.20 \cdot 10^{-5}$	$7.51 \cdot 10^{-5}$
10k evts	24h 40m	2d 11h	10d 15h	78d 1h

Numbers generated on dual 8-core Intel® Xeon® E5-2660 @ 2.20GHz

*,† Number of quarks limited to ≤6/4



Source: Schultz 2018

3 billions CPU hours/year
15% is MC integration

$$\sigma = \frac{1}{F} \int d\Phi |M|^2 \Theta(\Phi - \Phi_c)$$

phase-space factor — *integrand* — *phase-space cuts* — *probability distributions/ matrix element*

Agliardi, Grossi, Pellen, Prati "**Quantum integration of elementary particle processes.**" https://doi.org/10.1016/j.physletb.2022.137228

Michele Grossi plenary (yesterday)

QUANTUM TECHNOLOGY INITIATIVE — M. Grossi

Why quantum?

Fundamental scaling problems in generators:

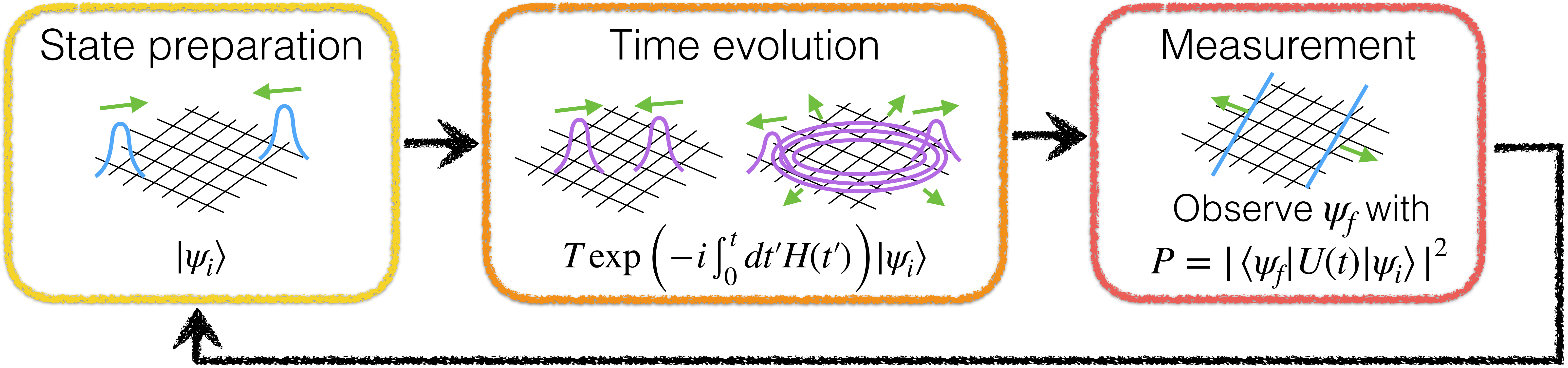
- Event complexity scales ~factorially with perturbation order
- Integration time scales ~exponentially with final-state multiplicity

Consequence of simulating a quantum system with classical computers

... because nature isn't classical, dammit, and if you want to ***make a simulation of nature****, you'd better* ***make it quantum mechanical****, ...*

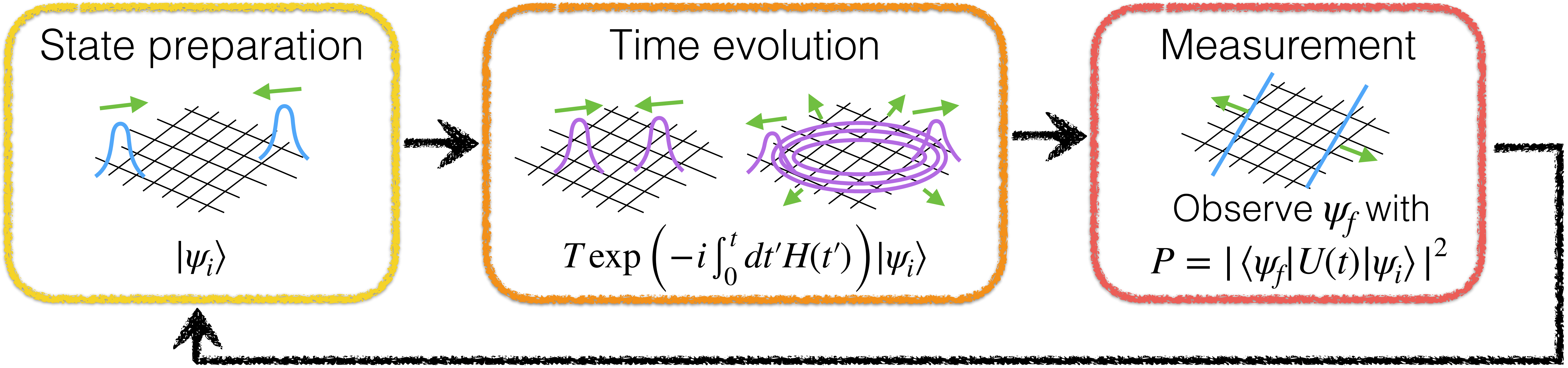
What a quantum event generator would look like

Real-time dynamics simulation + shot-by-shot sampling:

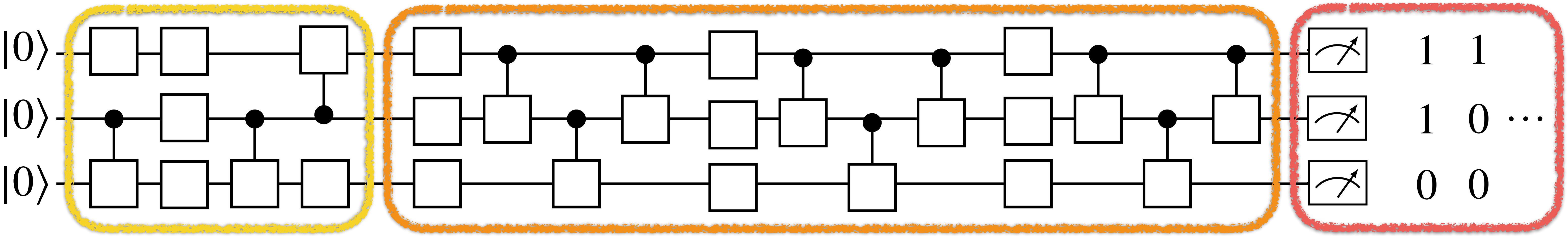


What a quantum event generator would look like

Real-time dynamics simulation + shot-by-shot sampling:



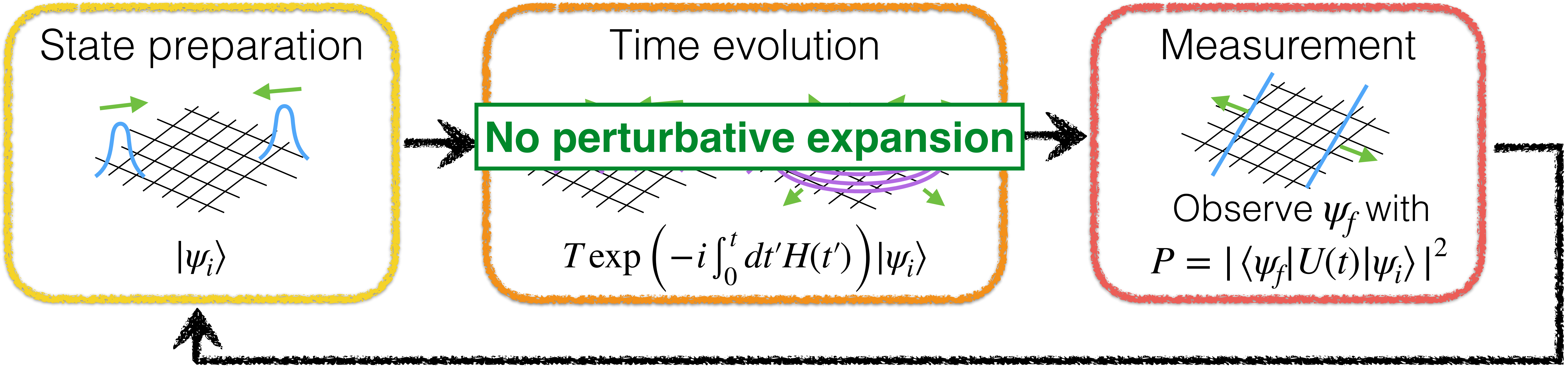
Which is, incidentally, how quantum computation works:



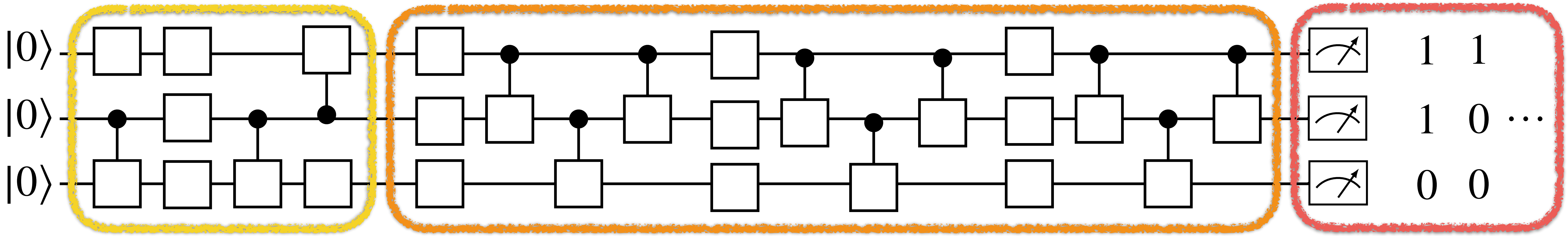
* If based on the quantum circuit model of quantum computing

What a quantum event generator would look like

Real-time dynamics simulation + shot-by-shot sampling:



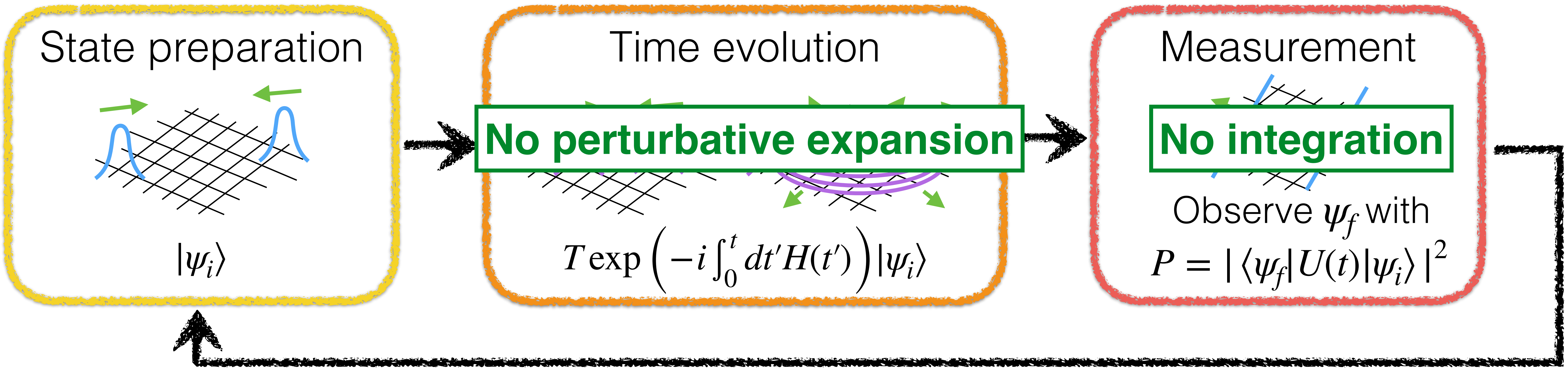
Which is, incidentally, how quantum computation works:



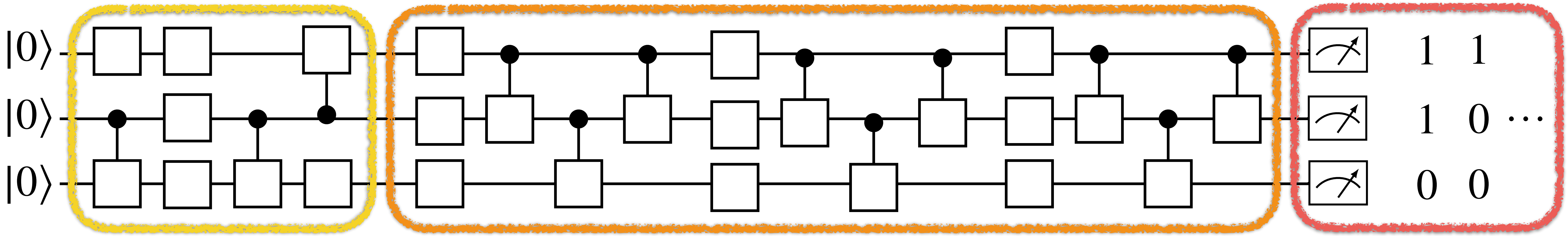
* If based on the quantum circuit model of quantum computing

What a quantum event generator would look like

Real-time dynamics simulation + shot-by-shot sampling:

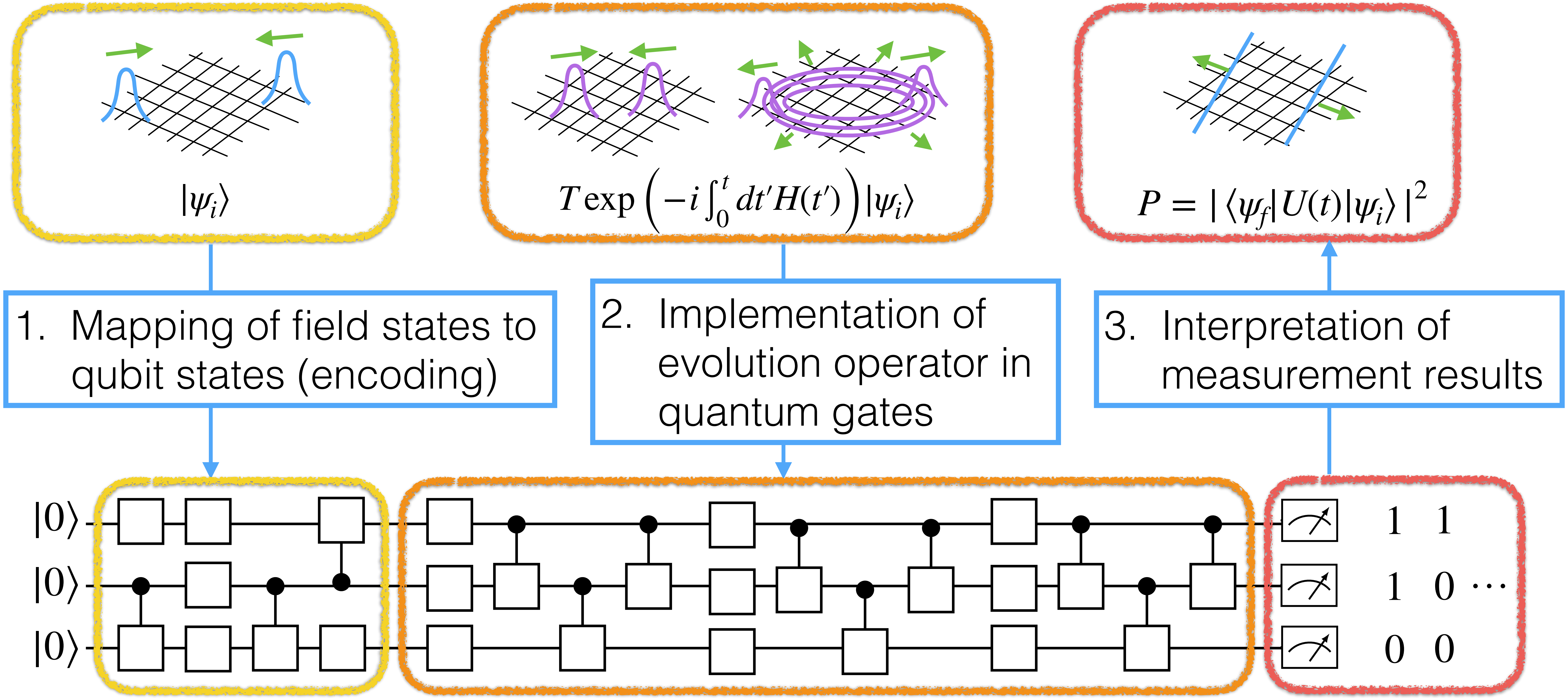


Which is, incidentally, how quantum computation works:



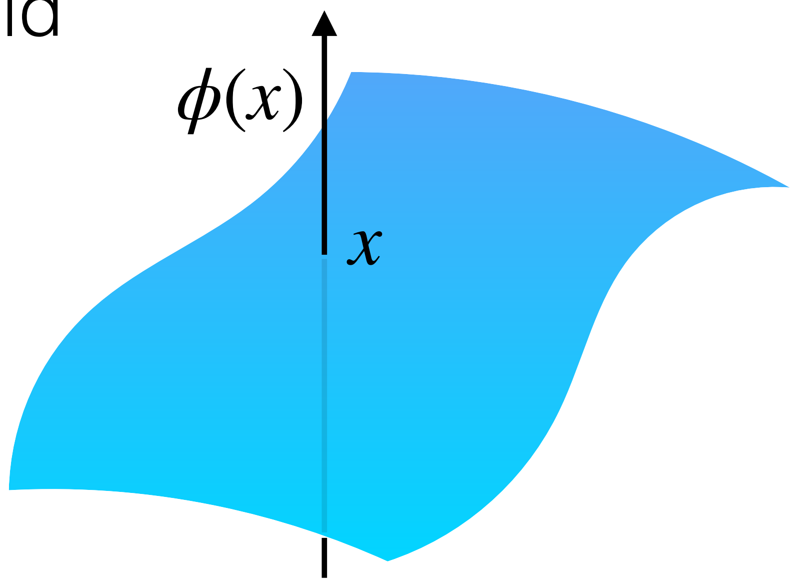
* If based on the quantum circuit model of quantum computing

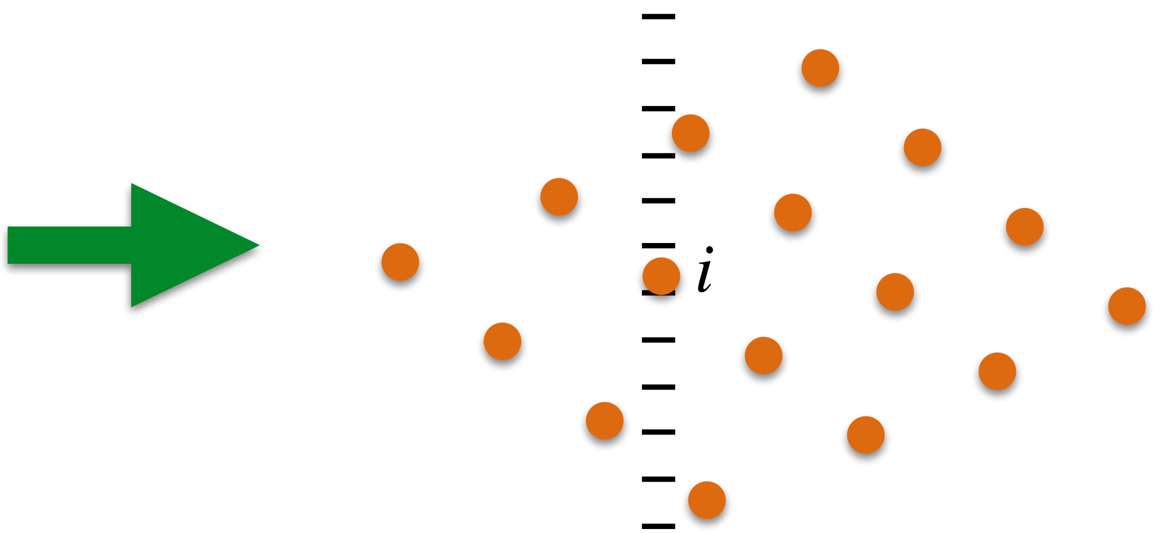
Ingredients of a quantum event generator



Encoding field states: discretization

Bosonic field





- Continuous (infinite) space $V = \int dx$
- Continuous unbounded field value ϕ

$$\Rightarrow \mathscr{H} = \text{span}\left(\left\{ |\phi\rangle \middle| \phi \in \mathbb{R} \right\}\right)^{\otimes \int dx}$$

$(\text{span}\left(\{|0\rangle, |1\rangle\}\right)$ for fermions)

- Discrete finite lattice $N = L^d$
- Discrete truncated field values $0,1,\ldots,K-1$

$$\Rightarrow \mathscr{H} = \text{span}\left(\{|0\rangle, |1\rangle, \ldots |K-1\rangle\}\right)^{\otimes N}$$

Discretization parameters determine the expressible dynamic range:

- $p_{\max}/p_{\min} \sim L$
- $\phi_{\max}/\phi_{\min} \sim K$

Field-based encoding

Use an n-bit quantum register per lattice point per field:

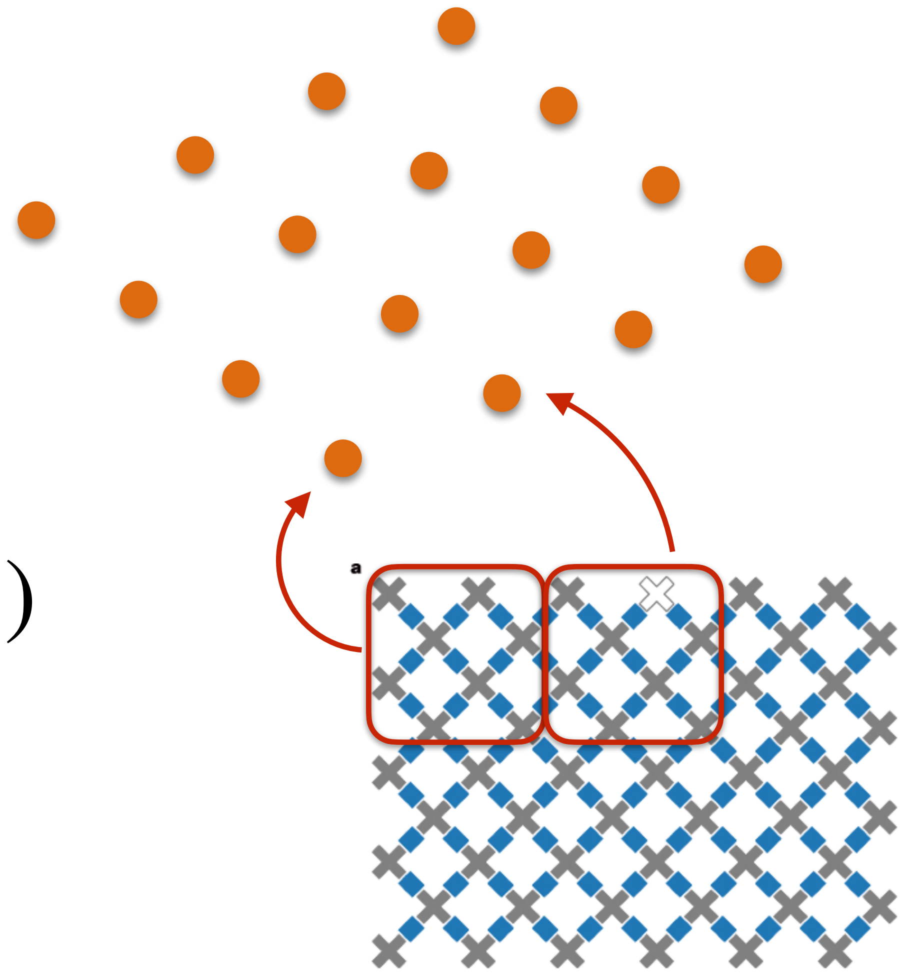
$|\text{system}\rangle = |j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_N\rangle \;\; (j_i = 0,\ldots,2^n - 1)$

Field value at site 1

Can also encode a Fock representation:

$|\text{system}\rangle = |k_{p_1}\rangle \otimes |k_{p_2}\rangle \otimes \cdots \otimes |k_{p_N}\rangle \;\; (k_{p_i} = 0,\ldots,2^n - 1)$

Number of excitations of mode p_1



⇒ **Qubit count**: nL^d

For $p_{\max}/p_{\min}$ = 10 TeV / 100 MeV = 10^5 and d = 3 we need ~$10^{15}n$ qubits

Alternative: Particle-based encoding

Assign a quantum register to each particle, maximum M particles

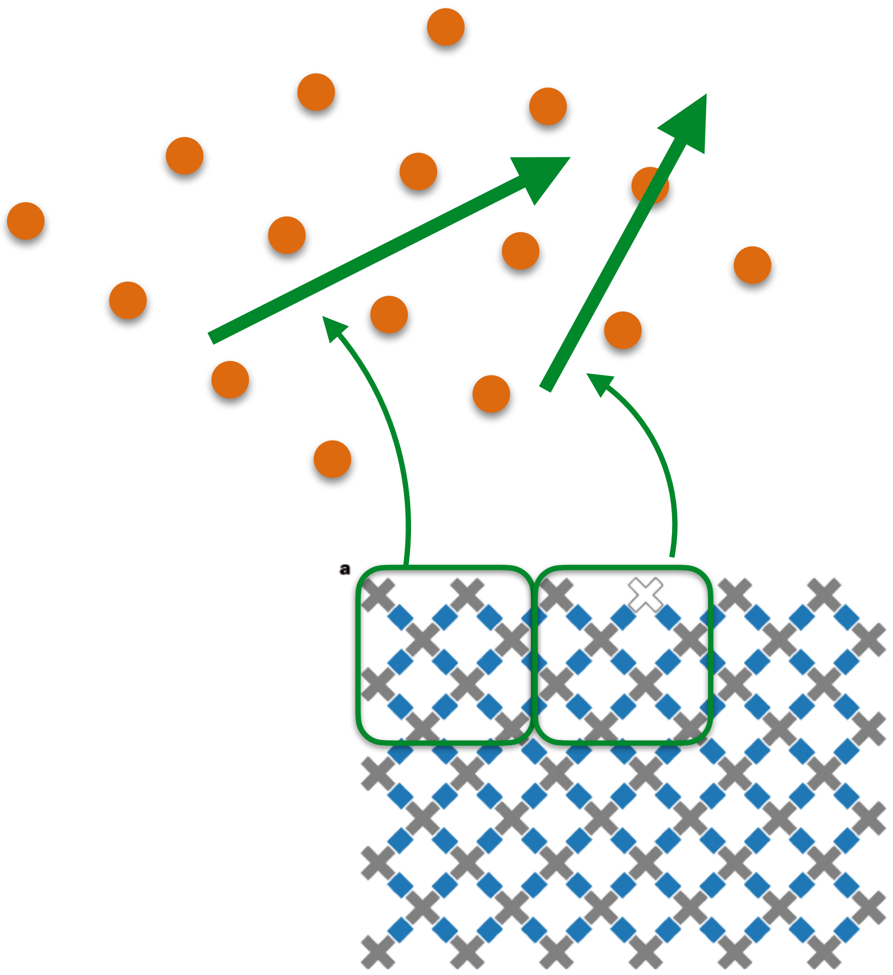
→ Field theory as multi-body quantum mechanics

$$|\text{system}\rangle = \mathcal{S}|p_1\ldots p_J\,\Omega\ldots\Omega\rangle$$

J occupied slots M-J unoccupied slots

Symmetrization (bosons) or antisymmetrization (fermions)

→ Slater determinant



⇒ **Qubit count**: $M(d\log_2 L)$

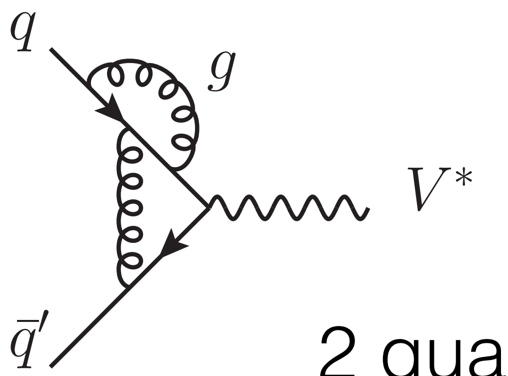
For p_{max}/p_{min} = 10^5 and d = 3 we need ~ $50M$ qubits

How many particles do we need?

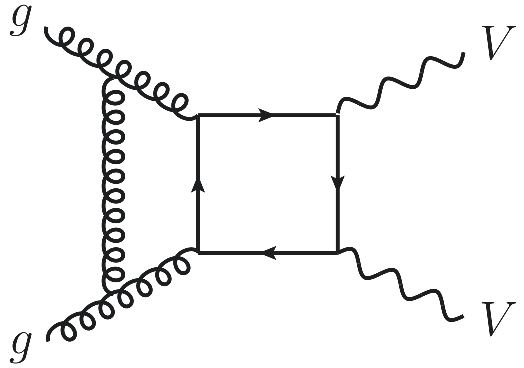
Diagram lines = particles.

For weak coupling, $\boldsymbol{M}$ ~ O(order of equivalent perturbative calculation)

Example:



2 quarks, 2 gluons, 1 V



2 quarks, 3 gluons, 2 V

→ Can include N^{x}LO contributions with moderate $\boldsymbol{M}$

Strong coupling: No prescription. Check convergence of observables as $\boldsymbol{M} \to \infty$

Constructing field operators

$$a_p \mathcal{S}|p_1 \ldots p \ldots p_J \,\Omega \ldots \Omega\rangle = \sqrt{n_p}\, \mathcal{S}|p_1 \ldots p_J \,\Omega\Omega \ldots \Omega\rangle$$

Annihilation operator de-occupies one slot..

$$a_q \mathcal{S}|p_1 \ldots p_J \,\Omega \ldots \Omega\rangle = 0 \quad (q \notin \{p_j\}_j)$$

or annihilates the ket if no matching occupied slot exists.

$$a_q^\dagger \mathcal{S}|p_1 \ldots p_J \,\Omega\Omega \ldots \Omega\rangle = \sqrt{n_q + 1}\, \mathcal{S}|p_1 \ldots p_J q \,\Omega \ldots \Omega\rangle$$

Creation operator fills one slot..

$$a_q^\dagger \mathcal{S}|p_1 \ldots p_M\rangle = 0$$

or annihilates the ket if it is maximally filled.

All operators can be expressed with combinations of a and $a^\dagger$

⇒ Figure out the implementation of $\mathcal{S}$, a, and $a^\dagger$!

Proposed implementations

- Barata et al. (PRA 103, 2021)

Only for bosons

$$\mathcal{S}|p_1 \ldots p_J \,\Omega \ldots \Omega\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{P \in \text{perm}(M)} |P(p_1 \ldots p_J \,\Omega \ldots \Omega)\rangle$$

$a_p^\dagger = \frac{1}{\sqrt{M}} \sum_j a_p^{\dagger(j)}$ where $a_p^{\dagger(j)}$ creates a particle in register j

- Gálves-Viruet and Llanes-Estrada (arXiv 2406.03147)

$$\mathcal{S}|p_1 \ldots p_J \,\Omega \ldots \Omega\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{P \in \text{perm}(J)} \sigma_P |P(p_1 \ldots p_J)\,\Omega \ldots \Omega\rangle$$

Sign of P

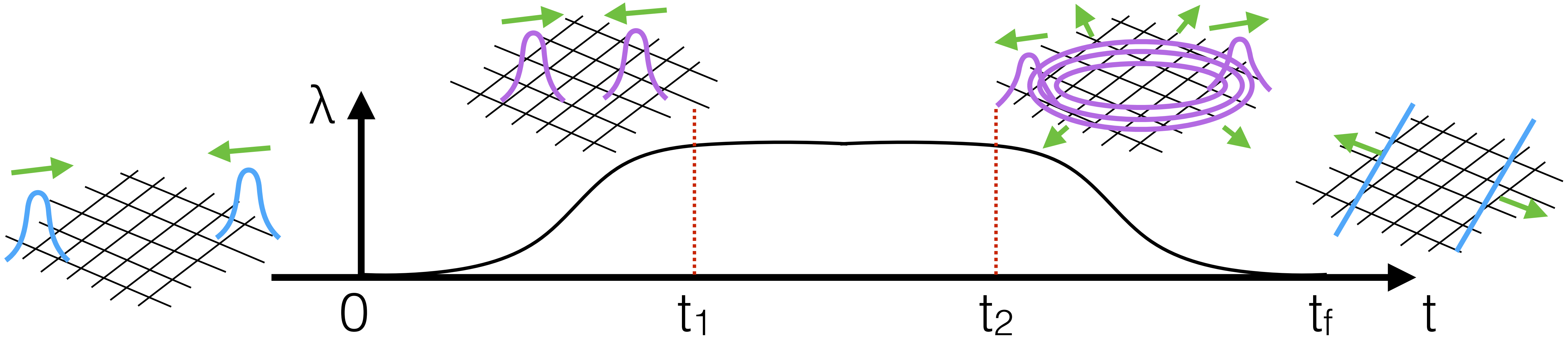
$a_p^\dagger = \sum_j \mathcal{T}_{j \leftarrow (j-1)}\, a_p^{\dagger(j)}$ where $a_p^{\dagger(j)}$ creates a particle in register j

and $\mathcal{T}_{j \leftarrow (j-1)}$ is a “step (anti)symmetrizer”

Event synopsis

Details in Barata et al.

- State preparation = Create wave packets $\sum_{\mathbf{p}_0,\mathbf{p}_1} \Psi_0(\mathbf{p}_0)\Psi_1(\mathbf{p}_1)\mathcal{S}|\mathbf{p}_0\mathbf{p}_1\,\Omega\ldots\Omega\rangle$
- Evolution in three time windows
  - $0 < t < t_1$: Adiabatic transition to physical single-particle states
    $H(t) = H_0 + f(t)\,H_I$ with $f(0) = 0, f(t_1) = 1$
  - $t_1 < t < t_2$: Evolution with full Hamiltonian e^{-iHt} (scattering)
  - $t_2 < t < t_f$: Adiabatic transition to Fock final states
- Measurement → Each bit string corresponds to a Fock state



Hamiltonian simulation

- Suzuki-Trotter decomposition (product formula)

$$\Delta t = t_f / N_{\text{step}}$$

$$\exp\left(-i \textstyle\sum_k H_k \Delta t\right) = \textstyle\prod_k \exp\left(-i H_k \Delta t\right) + \mathcal{O}\left((\varepsilon \Delta t)^2\right)$$

Full Hamiltonian is very complex
→ No quantum gate corresponding to $e^{-iH\Delta t}$

Decompose into small parts implementable with gates

Repeat Δt evolution for N_{step} times

- Block encoding of H + quantum signal processing

Embedding a non-unitary matrix in a larger unitary

Computation of a broad range of polynomials $f(x)$ for a given $x \in [-1,1]$

$$\begin{pmatrix} H & * \\ * & * \end{pmatrix} \longrightarrow \begin{pmatrix} c(Ht) - is(Ht) & * \\ * & * \end{pmatrix}$$

c, s: polynomial approximations of cos & sin

Biggest challenges

- Optimality of the encoding?

  Symmetrizers are complex & non-unitary. Any way around?

- How do we encode gauge symmetry?

  Gauge theory is not written in the language of particles..

- How do we select final states?

  A faithful LHC simulation will generate uninteresting events 99.999% of the time

- Circuit depth

  Interaction Hamiltonian requires $O(L^d)$ gates per time step / poly degree

Dev tool for particle-based quantization

```python
universe = Universe(
    [
        FieldDefinition(r'\phi', spin=0, max_particles=4),
        FieldDefinition(r'\psi', spin=1, max_particles=4, quantum_numbers=[('Charge', 2)])
    ],
    spatial_dimension=2
)
```
✓ 0.0s Python

```python
universe.null_state()
```
✓ 0.0s Python

$$[|\Omega\rangle \otimes |\Omega\rangle \otimes |\Omega\rangle \otimes |\Omega\rangle]_{\psi} \otimes [|\Omega\rangle \otimes |\Omega\rangle \otimes |\Omega\rangle \otimes |\Omega\rangle]_{\phi}$$

```python
field = universe.fields[r'\psi']

px, py, qx, qy = symbols('p_x, p_y, q_x, q_y', integral=True)

apply_op(
    field.creation_op((px, py), spin=1, Charge=-1)
    * field.creation_op((qx, qy), spin=-1, Charge=1)
    * field.null_state()
)
```
✓ 0.0s Python

$$-\frac{\sqrt{2}|\Omega\rangle \otimes |\Omega\rangle \otimes |q_x, q_y : -1; 1\rangle \otimes |p_x, p_y : 1; -1\rangle}{2} + \frac{\sqrt{2}|\Omega\rangle \otimes |\Omega\rangle \otimes |p_x, p_y : 1; -1\rangle \otimes |q_x, q_y : -1; 1\rangle}{2}$$

https://github.com/yiiyama/pb2q

Sympy-based toolkit for

- Algorithm dev & validation
- Numerical calculations
- Visualization

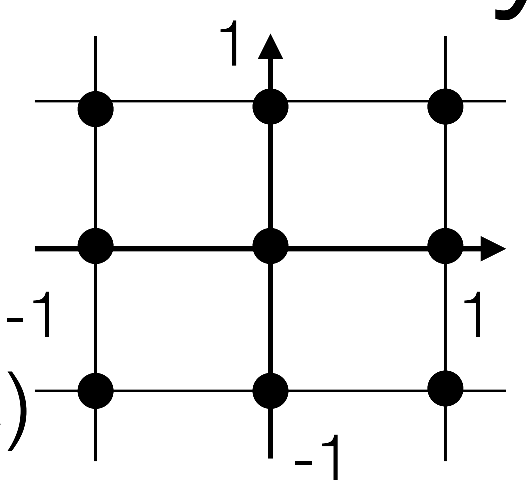
Demonstration: ground state of ϕ^4 theory

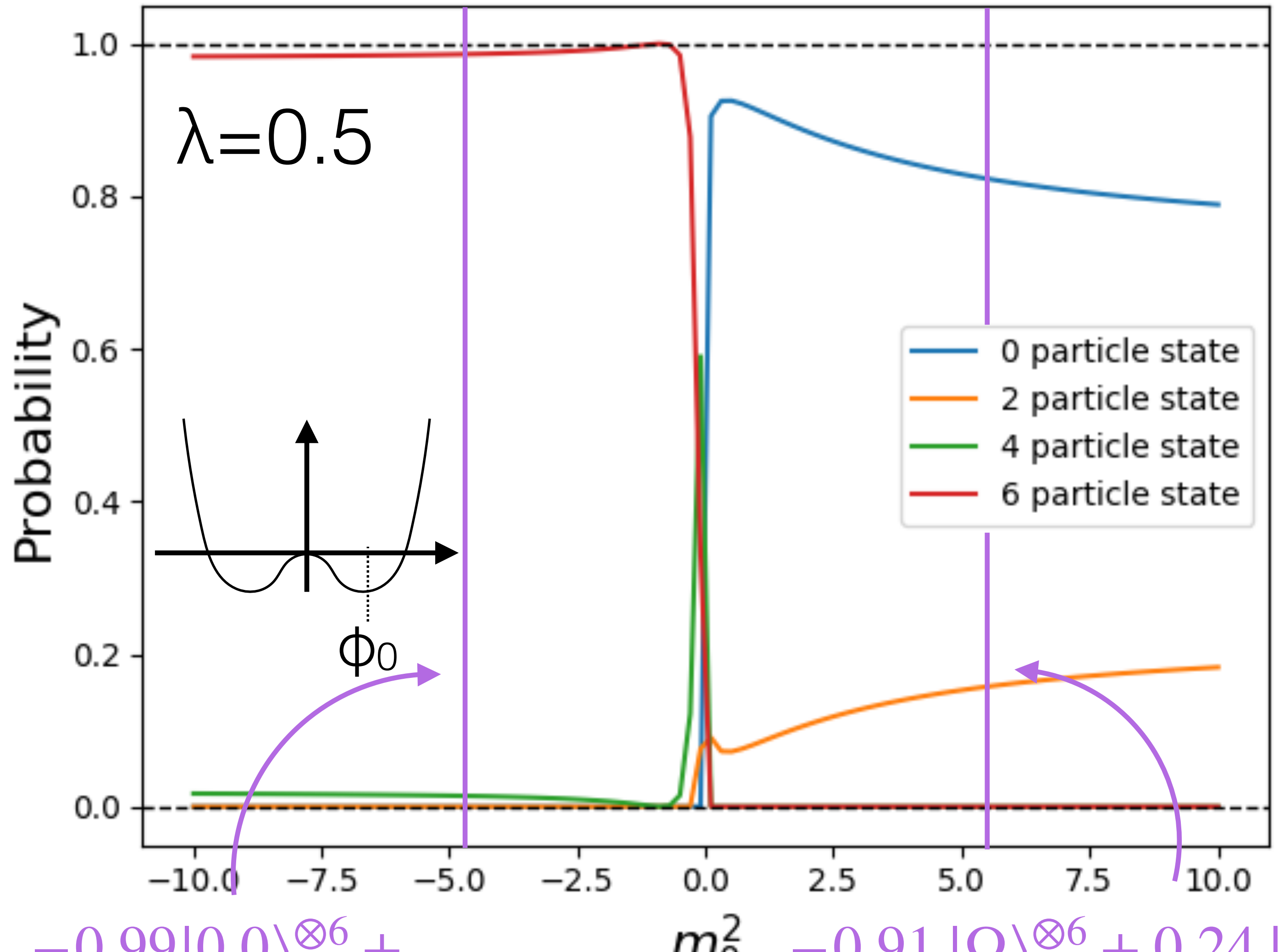
$$H = \frac{1}{2}\int d\mathbf{x}\left[\pi(\mathbf{x})^2 + \left(\nabla\phi(\mathbf{x})\right)^2 + m^2\phi(\mathbf{x})^2 + \delta_m\phi(\mathbf{x})^2 + \frac{\lambda}{12}\phi(\mathbf{x})^4\right]$$

discretized

$$H = \sum_{\mathbf{p}} \omega_{\mathbf{p}} a^\dagger_{\mathbf{p}} a_{\mathbf{p}} + \frac{1}{2}\sum_{\mathbf{n}}\left(\delta_m\phi_{\mathbf{n}}^2 + \frac{\lambda}{12}\phi_{\mathbf{n}}^4\right)$$

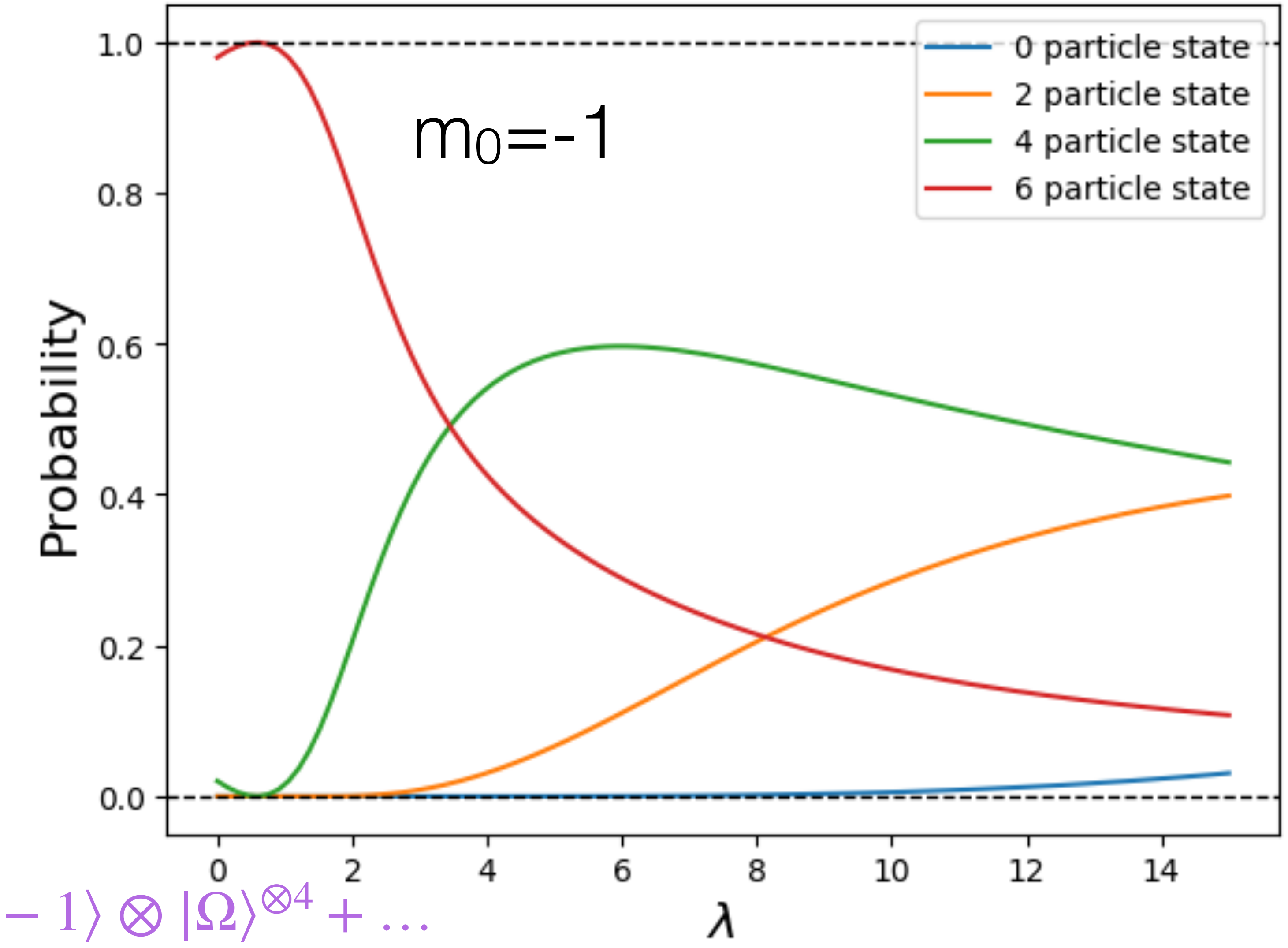
6 particles
9 2D momenta
$(M = 6, L = 3, d = 2)$





$-0.99|0,0\rangle^{\otimes 6} + \ldots$

$-0.91\,|\Omega\rangle^{\otimes 6} + 0.24\,|1,1\rangle \otimes |-1,-1\rangle \otimes |\Omega\rangle^{\otimes 4} + \ldots$



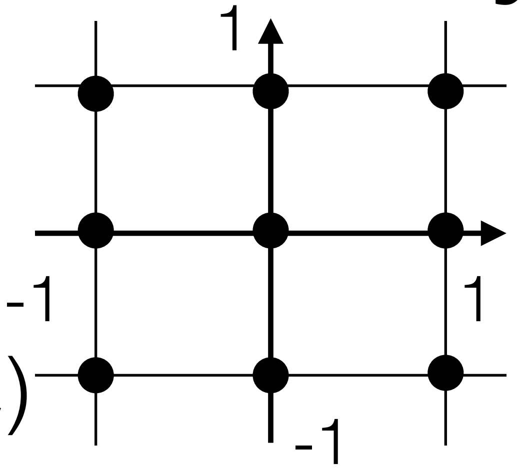
Demonstration: ground state of ϕ^4 theory

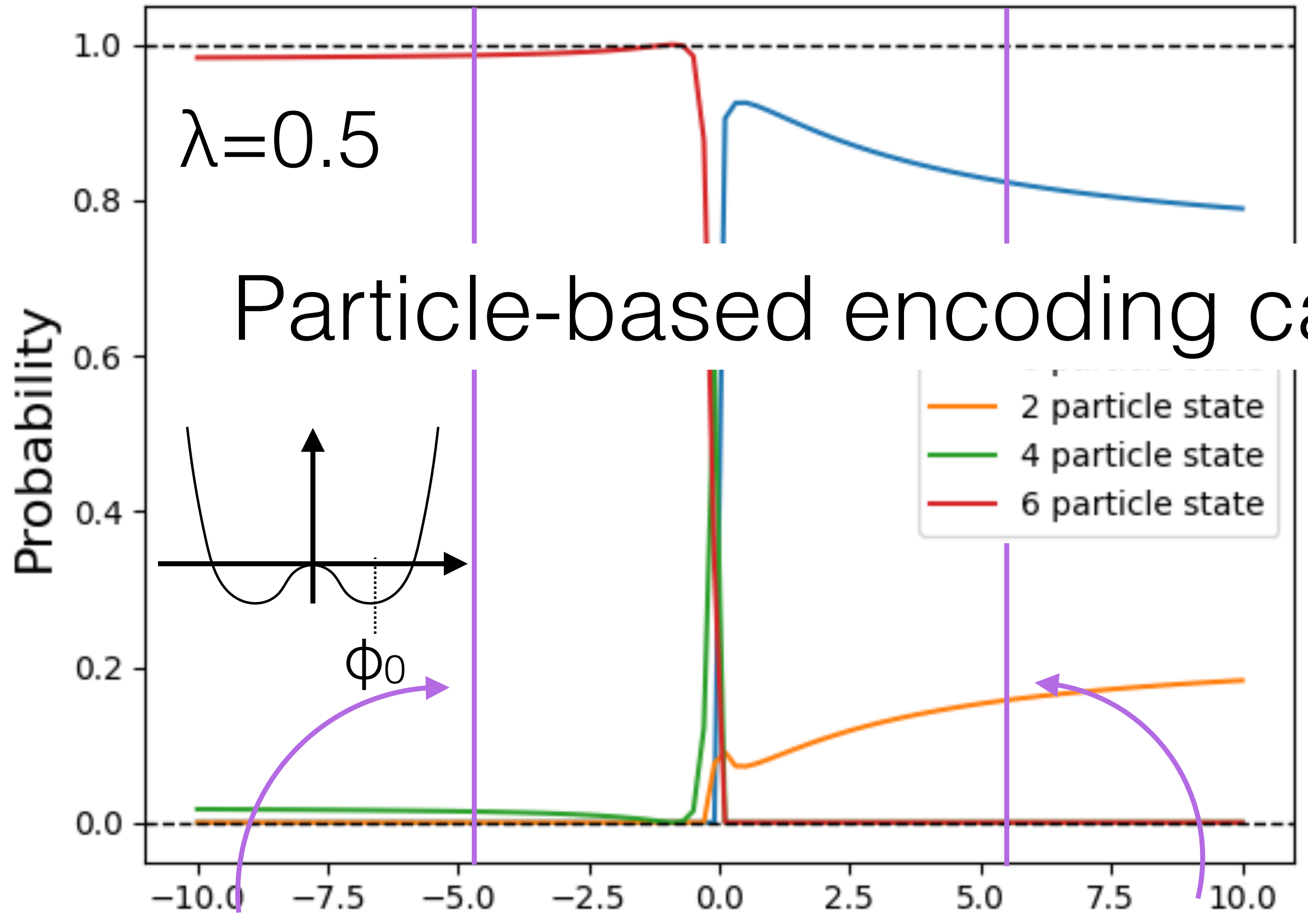
$$H = \frac{1}{2}\int d\mathbf{x}\left[\pi(\mathbf{x})^2 + \left(\nabla\phi(\mathbf{x})\right)^2 + m^2\phi(\mathbf{x})^2 + \delta_m\phi(\mathbf{x})^2 + \frac{\lambda}{12}\phi(\mathbf{x})^4\right]$$

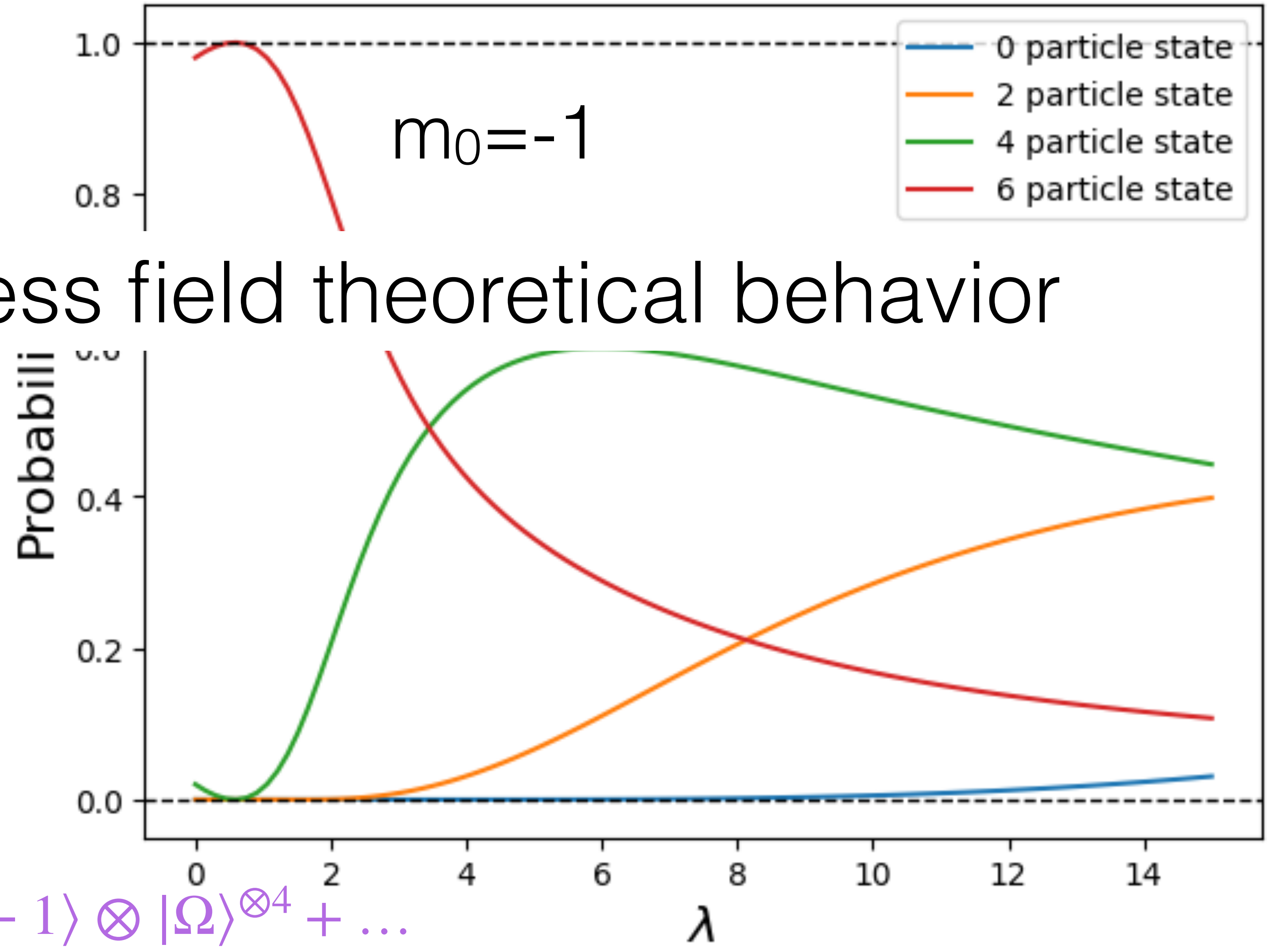
discretized

$$H = \sum_{\mathbf{p}}\omega_{\mathbf{p}}a^\dagger_{\mathbf{p}}a_{\mathbf{p}} + \frac{1}{2}\sum_{\mathbf{n}}\left(\delta_m\phi_{\mathbf{n}}^2 + \frac{\lambda}{12}\phi_{\mathbf{n}}^4\right)$$

6 particles
9 2D momenta
$(M = 6, L = 3, d = 2)$





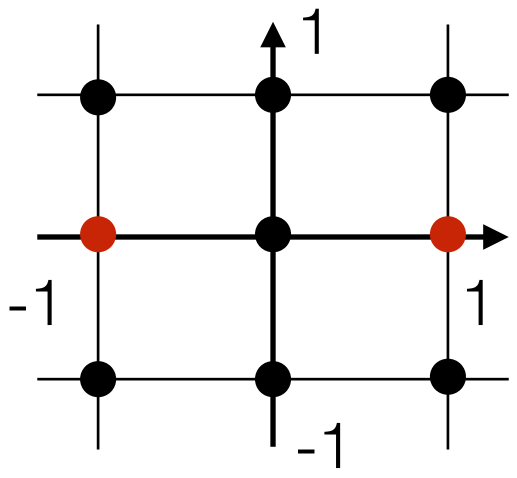


Particle-based encoding can express field theoretical behavior

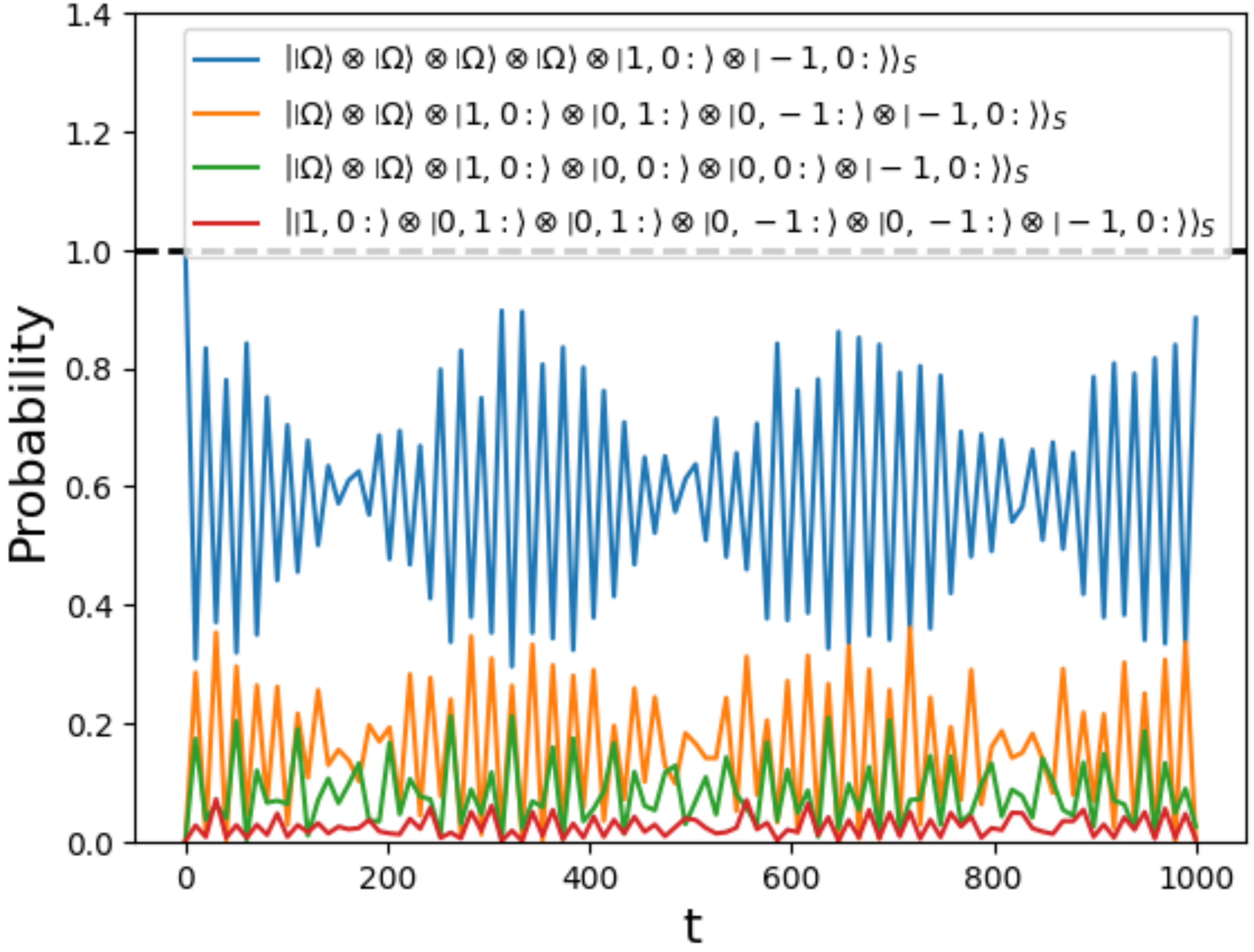
$-0.99|0,0\rangle^{\otimes 6} + \ldots$

$-0.91\,|\Omega\rangle^{\otimes 6} + 0.24\,|1,1\rangle \otimes |-1,-1\rangle \otimes |\Omega\rangle^{\otimes 4} + \ldots$

Demonstration: time evolution in ϕ^4



- Initial state $|1,0\rangle \otimes |-1,0\rangle \otimes |\Omega\rangle^{\otimes 4}$
- $\lambda=1$, $m_0=0.1$
- Not performing adiabatic turn on / off
  → Lattice too small to form wave packets



Conclusion

- We can use quantum computers for real-time simulation of quantum fields
- Evolve an initial state and measure → quantum event generator
    - Can emcompass N^xLO depending on truncation
    - No integration whatsoever
- Particle-based encoding uses realistic number of qubits
    - Is suitable for sparse problems → scattering
- Very early stage, still a lot to figure out
- **Let's build a quantum event generator together!**