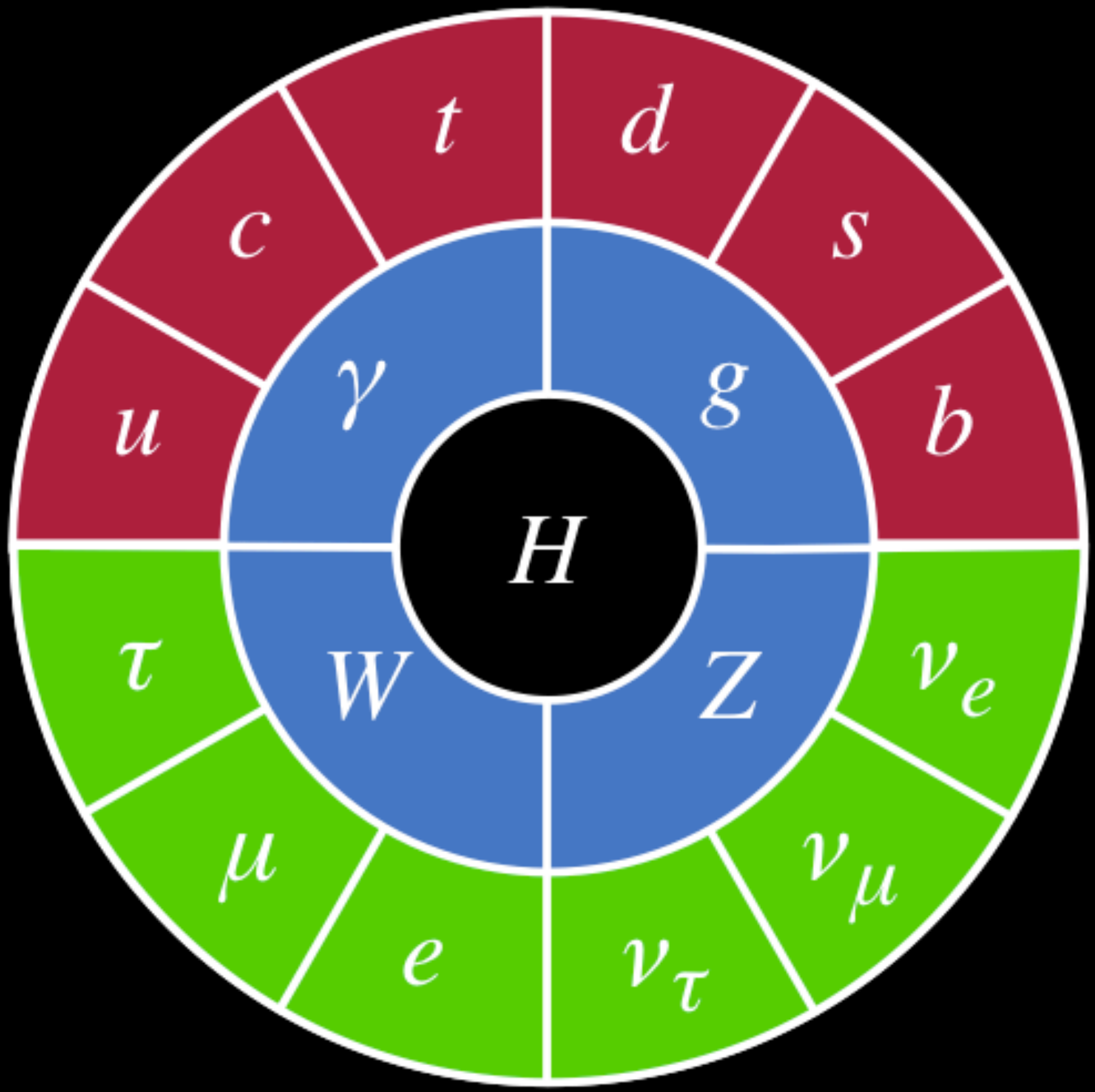
Quantum Algorithms for HEP simulations

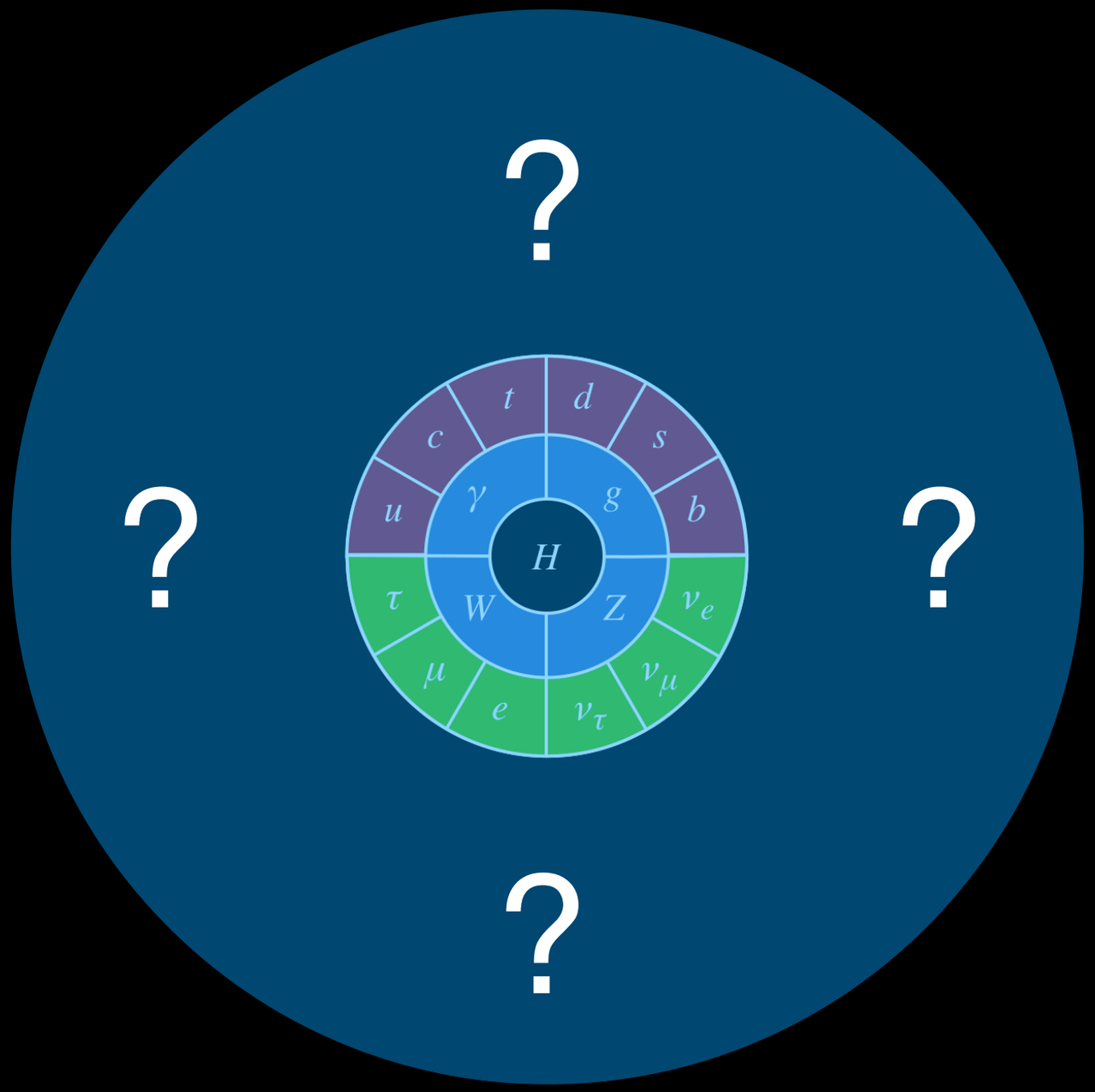
Quantum Information Science for Fundamental Physics, Aspen

CWB, W. de Jong, B. Nachman, D. Provasoli

1904.03196 [quant-ph]



Walter Murch,


?
?
?
?
t
d
c
s
u
b
γ
g
H
τ
W
Z
ν_e
μ
ν_μ
e
ν_τ

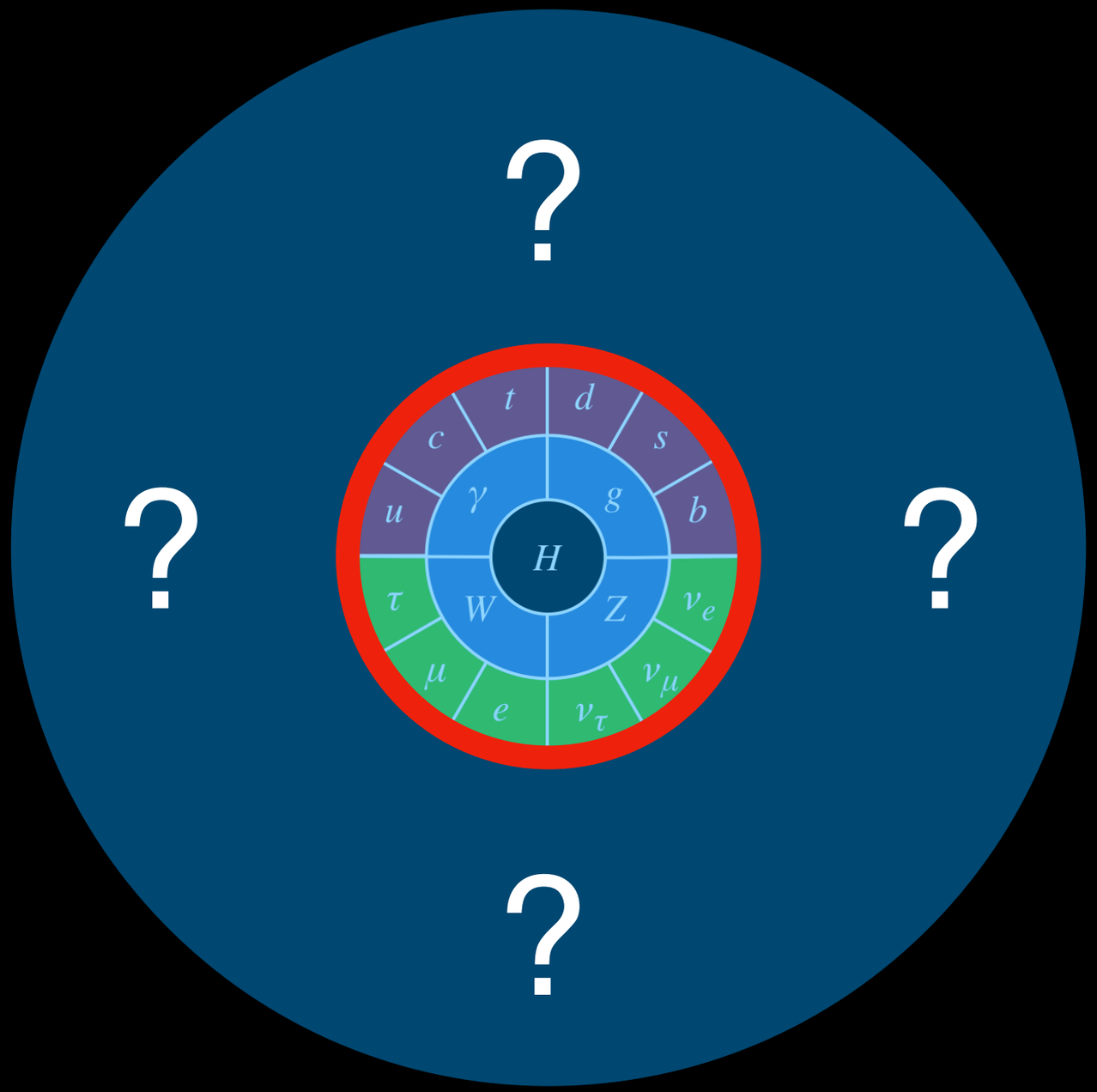
Christian Bauer

Quantum algorithms for High Energy Physics Simulations

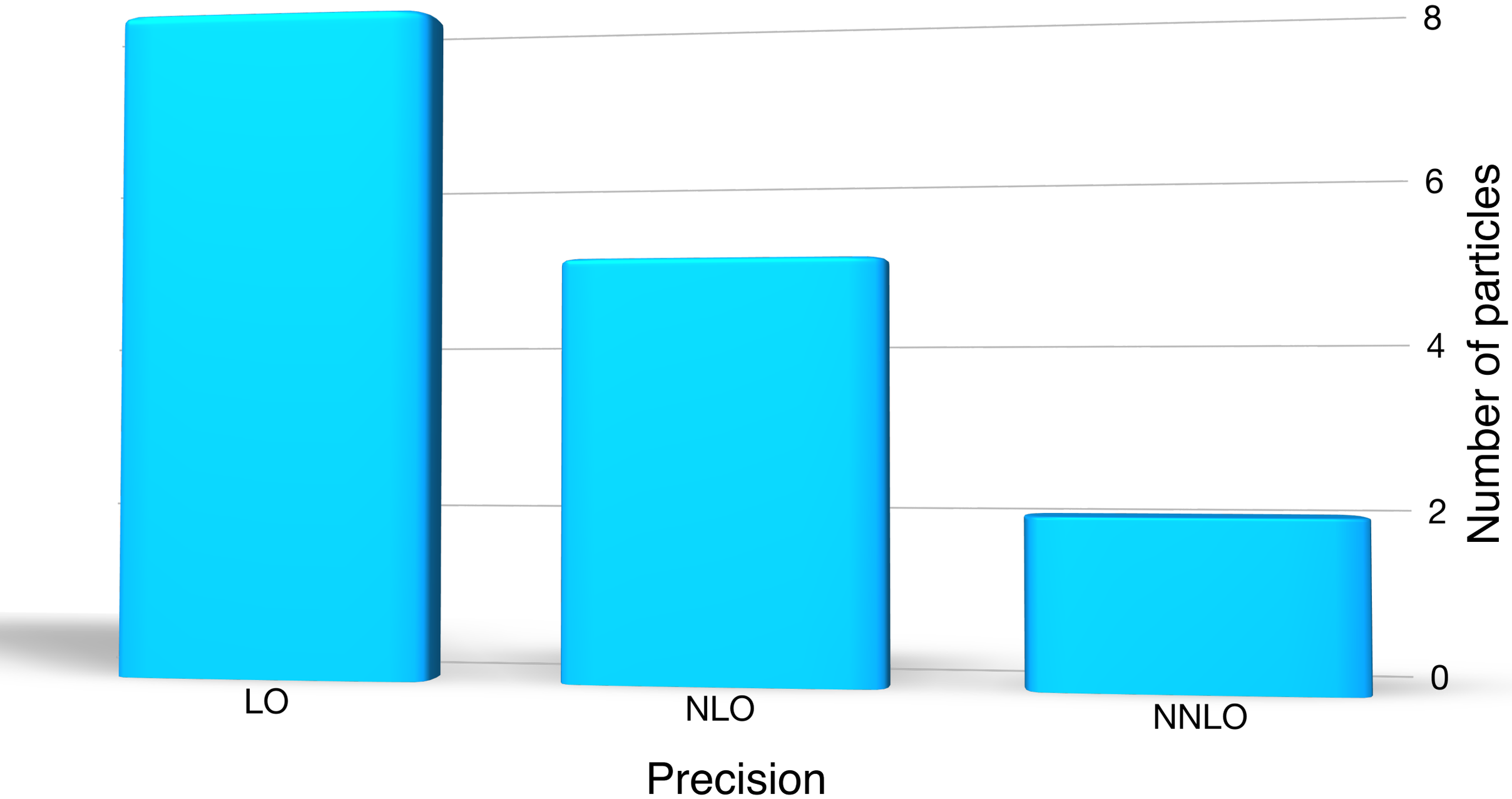
Christian Bauer

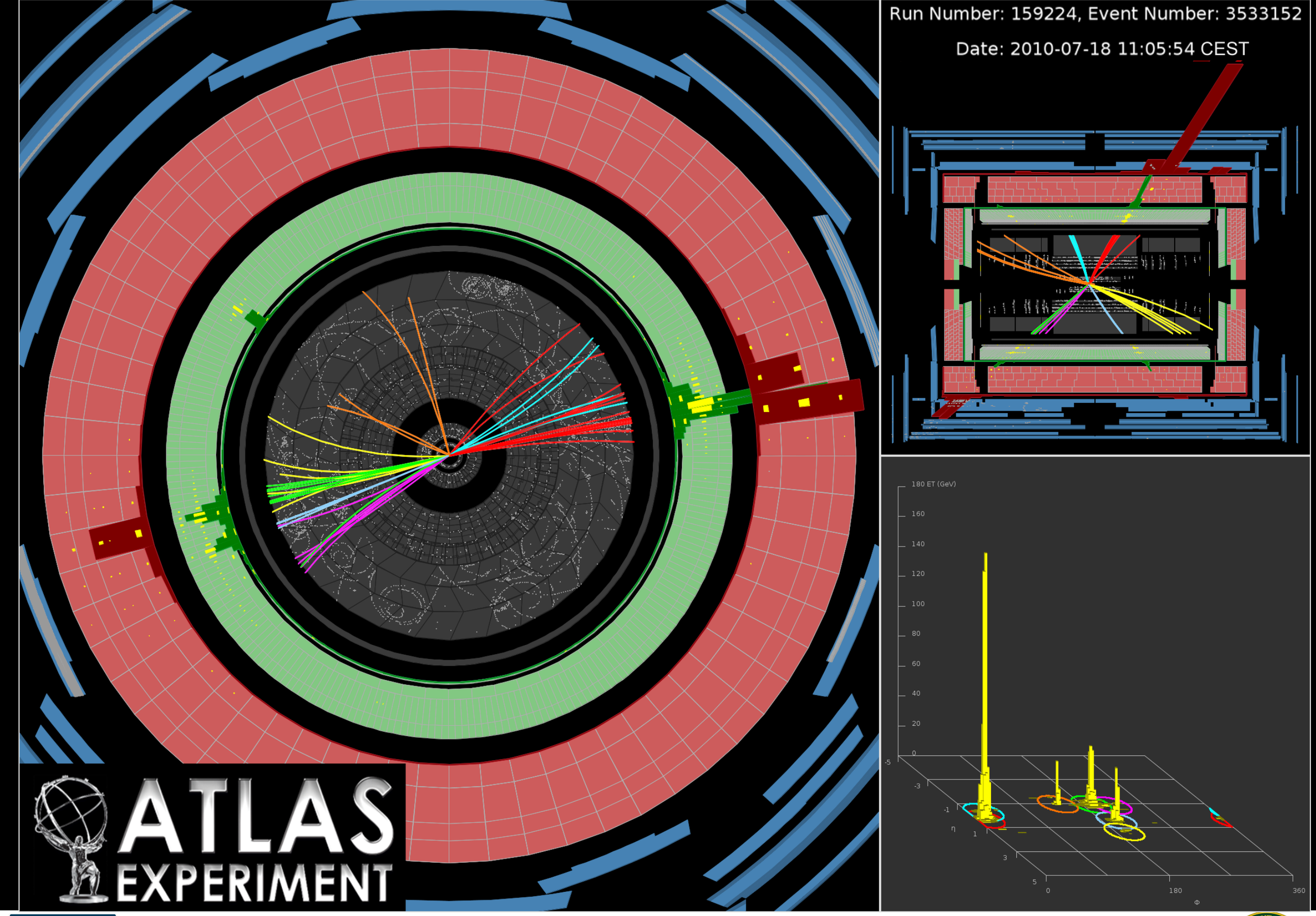
Quantum algorithms for High Energy Physics Simulations

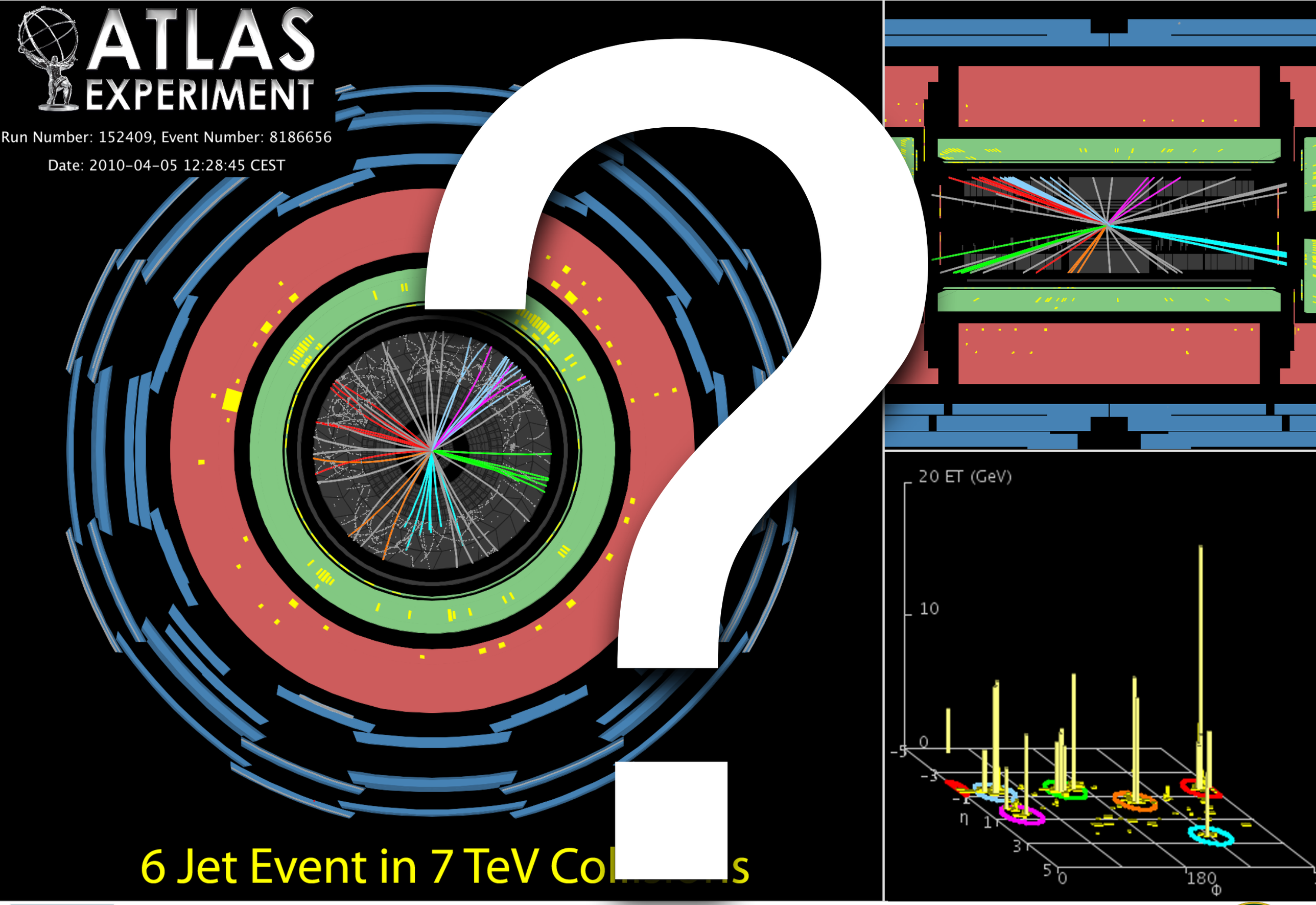
Need Precise predictions



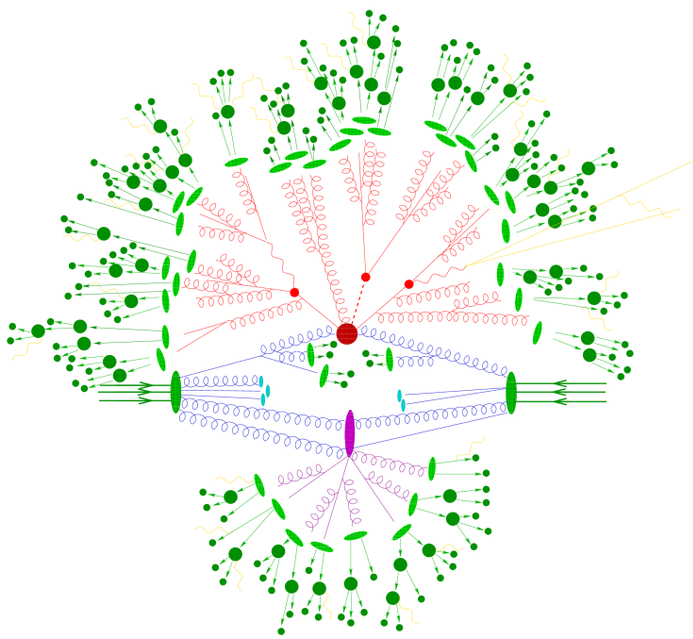
What we can compute is limited…



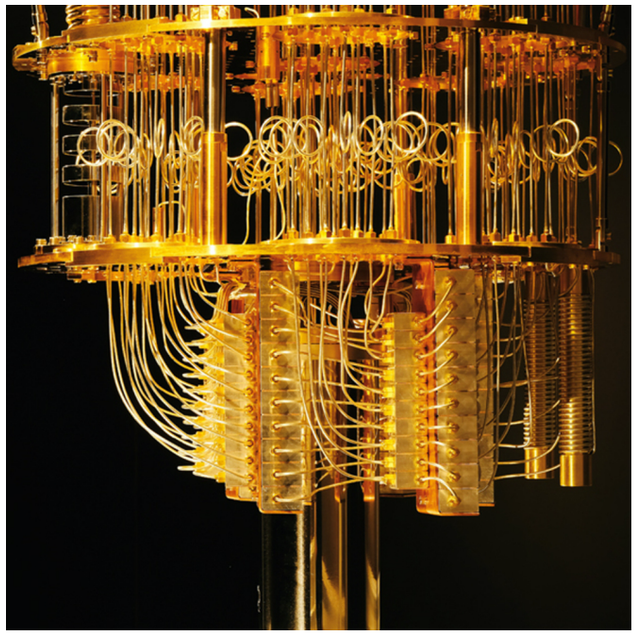

Run Number: 159224, Event Number: 3533152
Date: 2010-07-18 11:05:54 CEST
180 ET (GeV)
160
140
120
100
80
60
40
20
0
-5
-3
-1
η
1
3
5
0
180
360
Φ
ATLAS
EXPERIMENT



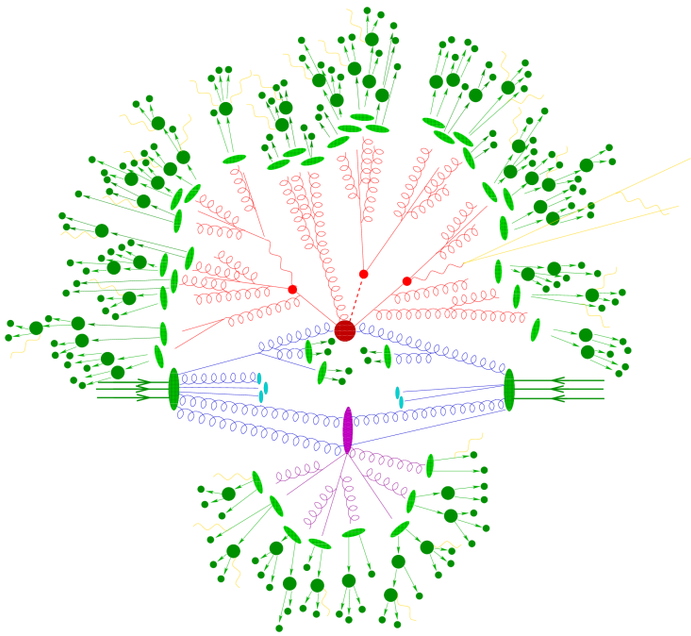
6 Jet Event in 7 TeV Co[illegible]s



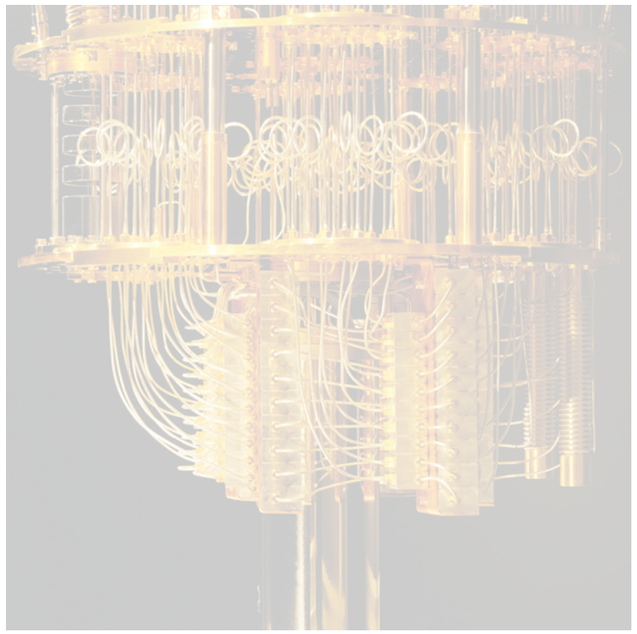
The traditional way to compute high multiplicity events is inherently probabilistic and can not include many quantum interference effects



Quantum algorithms have possibility to include such quantum effects efficiently

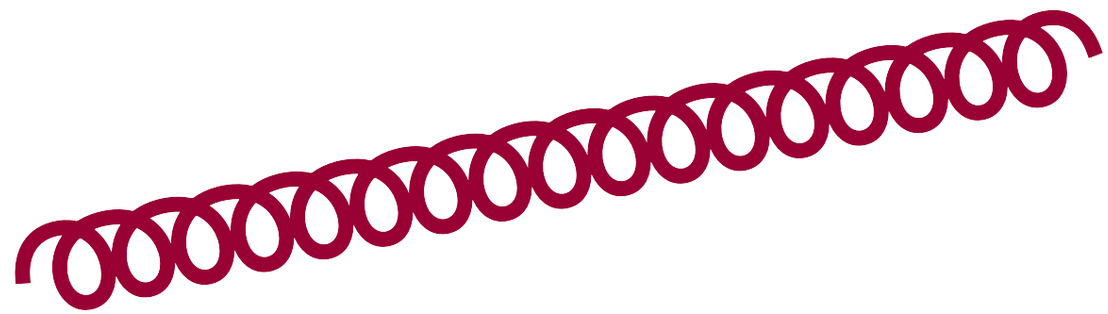


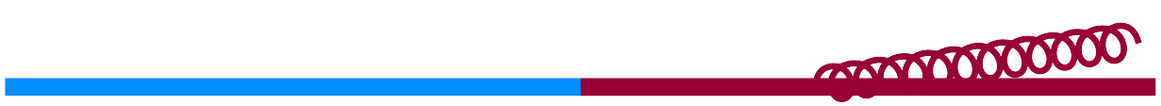
The traditional way to compute high multiplicity events is inherently probabilistic and can not include many quantum interference effects

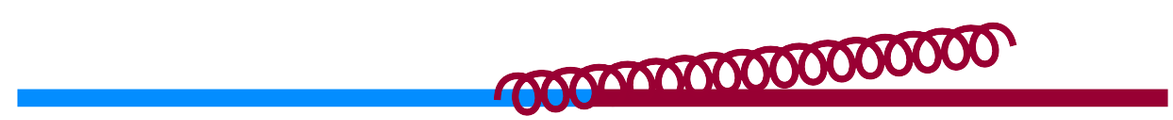


Quantum algorithms have possibility to include such quantum effects efficiently

A few very basic facts about amplitudes

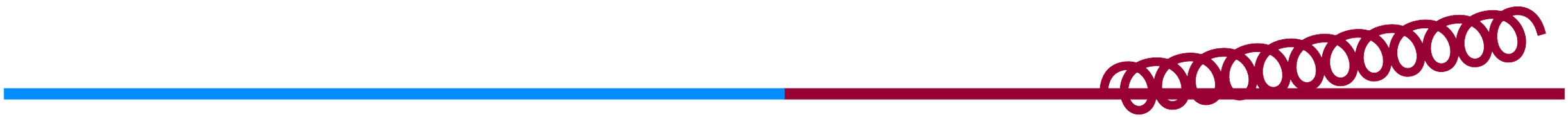






A few very basic facts about amplitudes

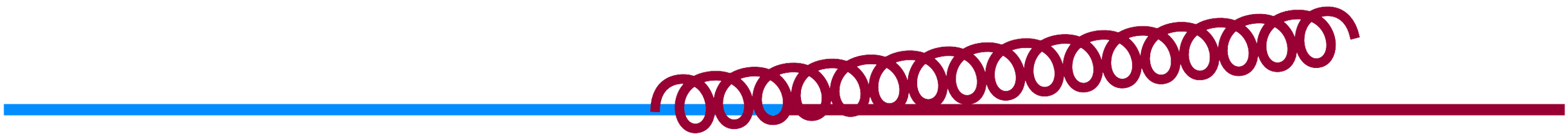
virtual particle almost real
likely process in QM
Amplitude large



$$A_q \sim \frac{1}{(p_q + p_g)^2}$$

A few very basic facts about amplitudes

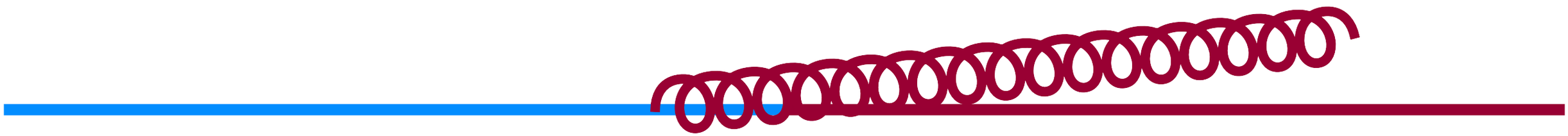
virtual particle far from real
unlikely process in QM
Amplitude small



$$A_q \sim \frac{1}{(p_{\bar{q}} + p_g)^2} \ll A_q \sim \frac{1}{(p_q + p_g)^2}$$

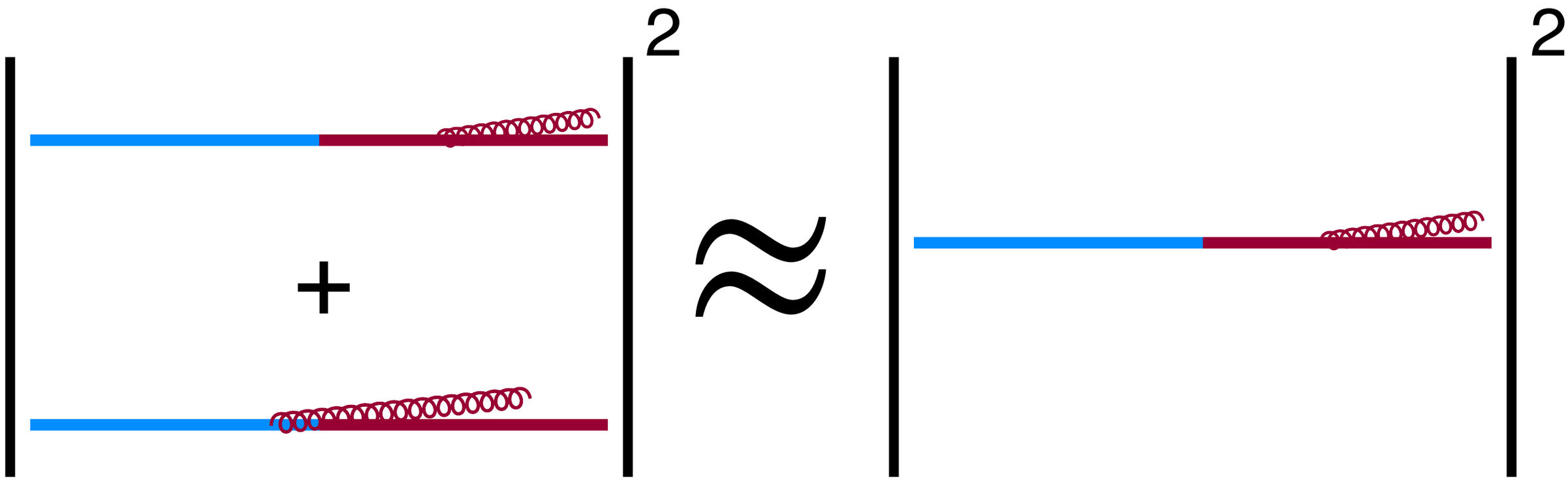
A few very basic facts about amplitudes

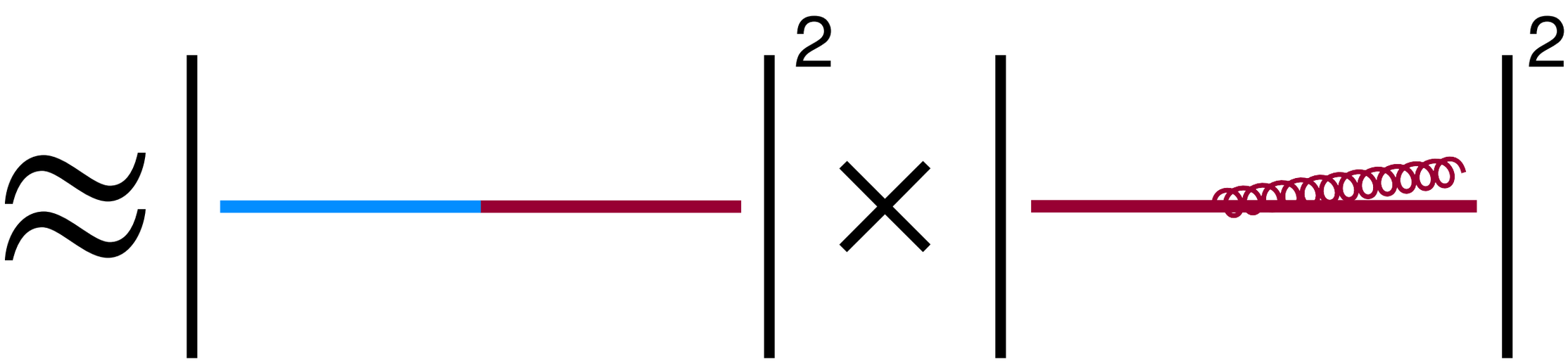
virtual particle far from real
unlikely process in QM
Amplitude small



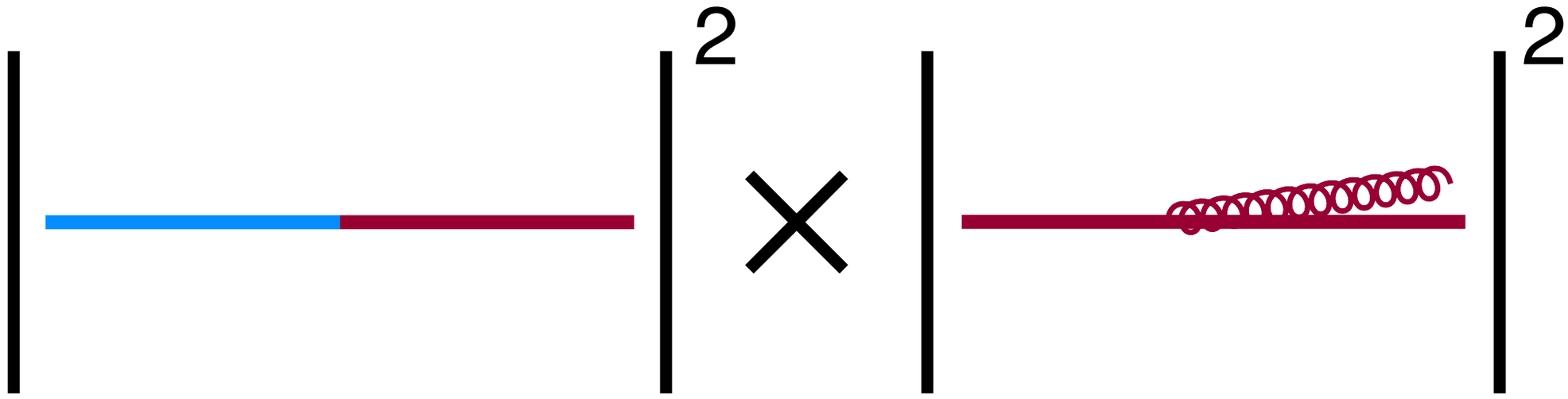
$$A_q \sim \frac{1}{(p_{\bar{q}} + p_g)^2} \ll A_q \sim \frac{1}{\underbrace{(p_q + p_g)^2}_{t}}$$

A few very basic facts about amplitudes





A few very basic facts about amplitudes



$$\left|A_{n+1}\right|^2 \approx \left|A_n\right|^2 \times P(t)$$

Dealing with probabilities instead of amplitudes

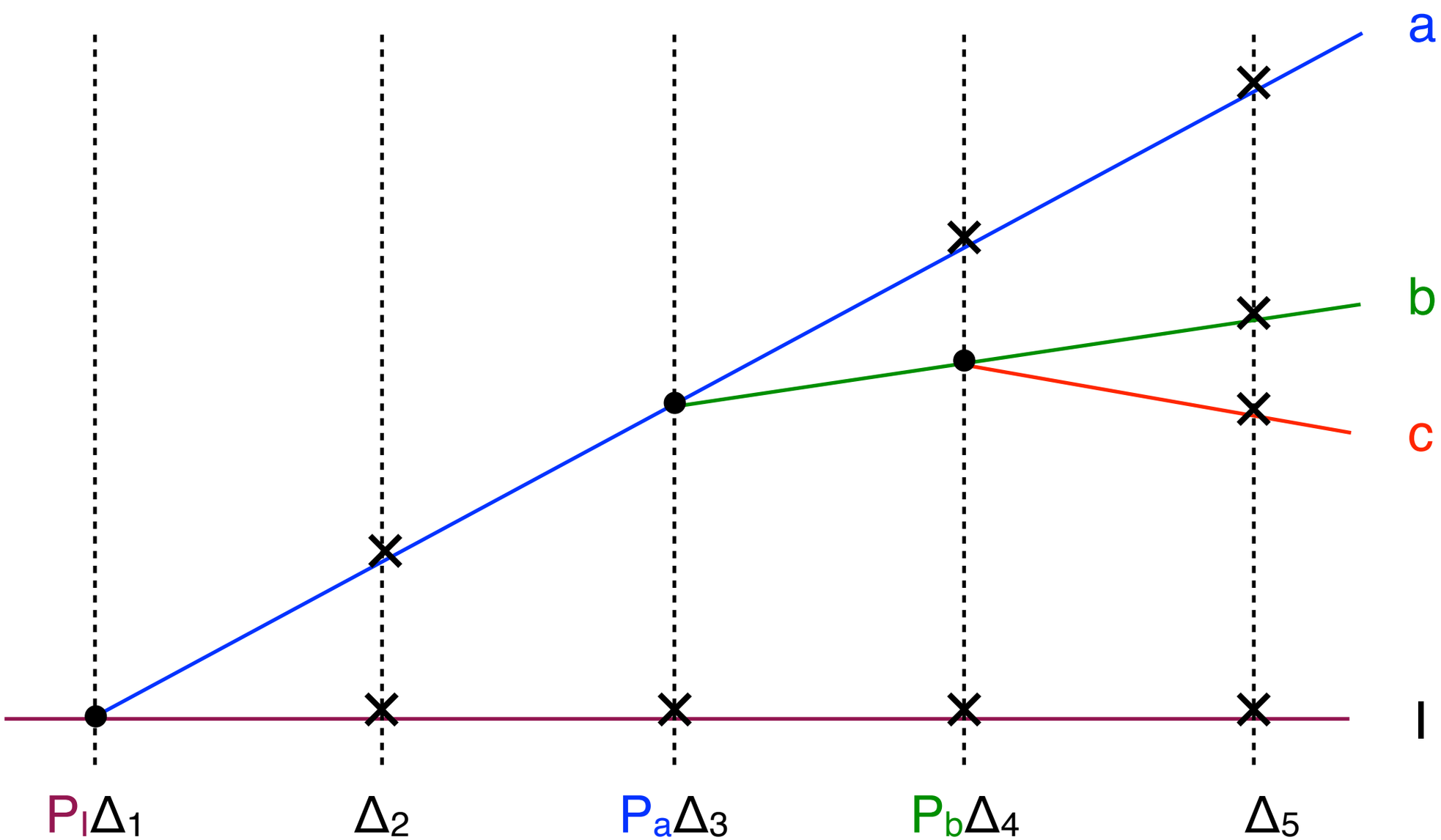
A few very basic facts about amplitudes

$$\left|A_{n+1}\right|^2 \approx \left|A_n\right|^2 \times P(t)$$

Whole problem Markovian process

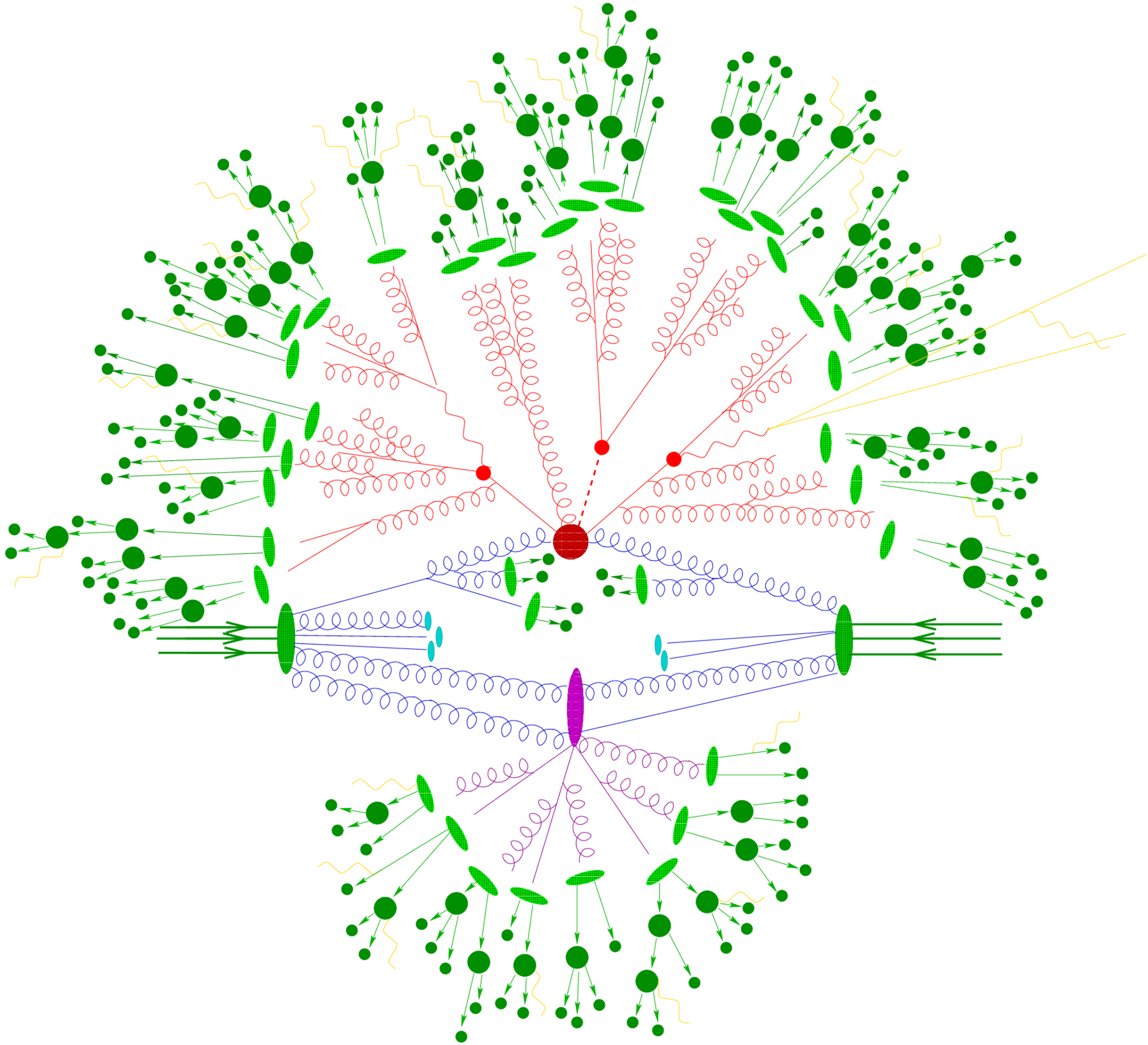
Two possibilities at each t:

1. Nothing happens (no-branch prob Δ)
2. Emission happens (branch prob $P \times \Delta$)



Emission depends on P of particle that emits and Δ of system at time t_i

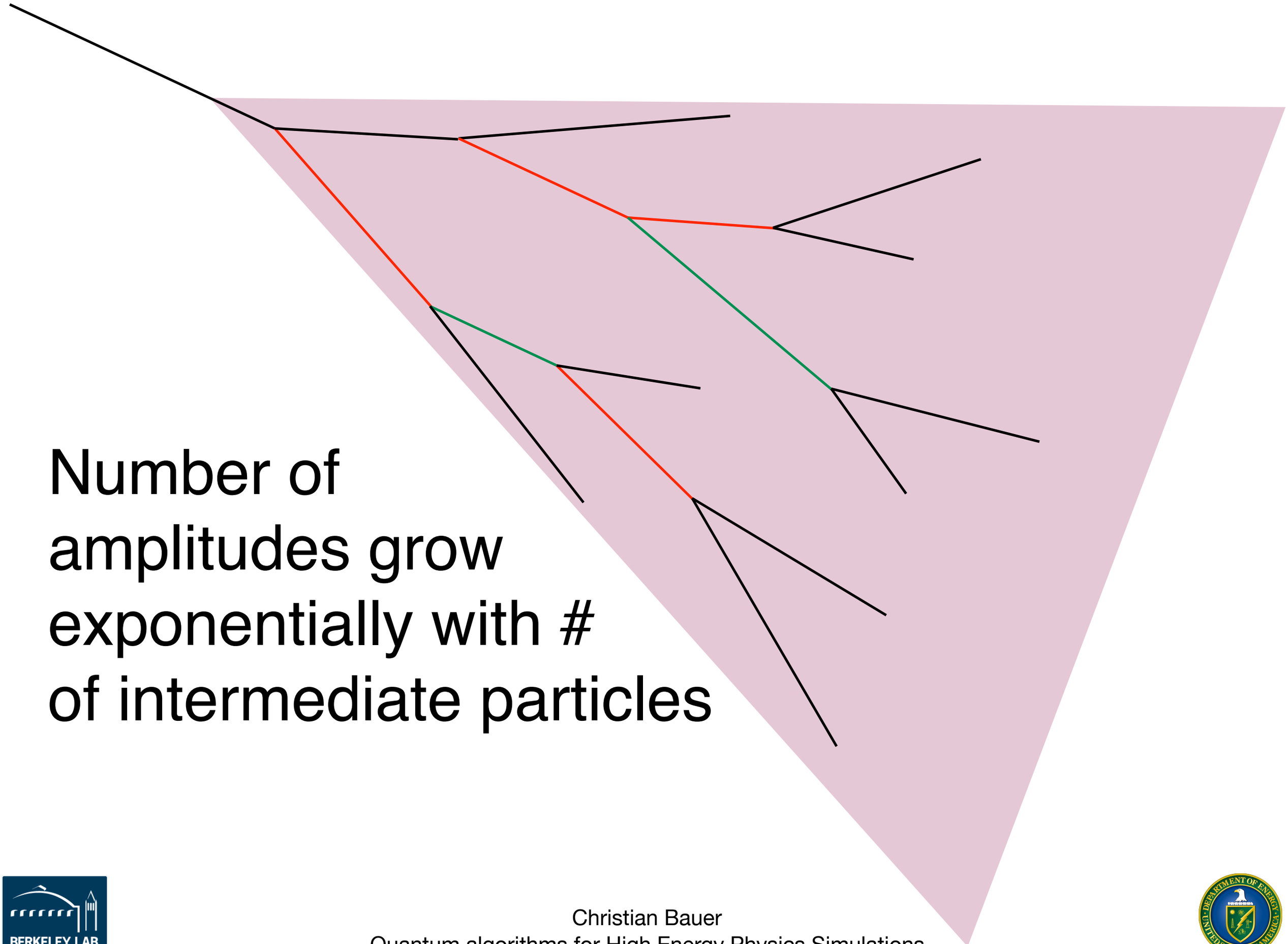
```
state = initial_state()
for t in 1... N:
  if emission_happens(state):
    n = choose_emitter(state)
    state = new_state(state, n)
write_out(state)
```



Quantum algorithms for High Energy Physics Simulations

…but parton shower is completely based on probabilities, so all quantum mechanical information is lost...

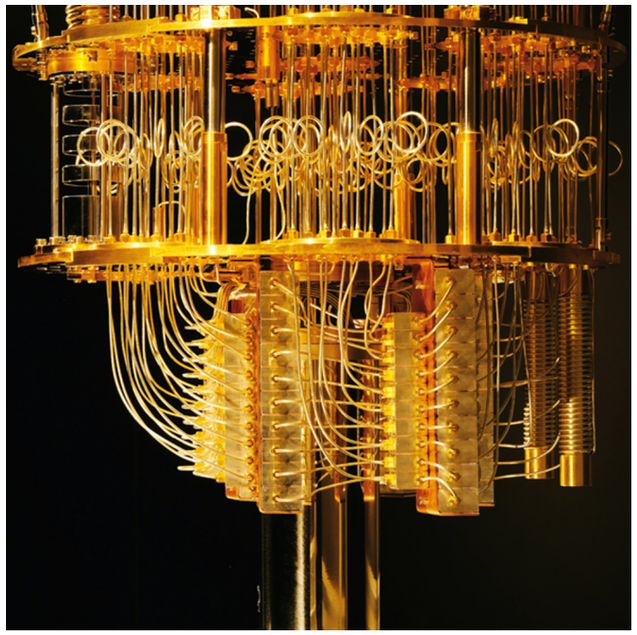
…to get it back, need to compute shower for each possible amplitude…



Number of amplitudes grow exponentially with # of intermediate particles

Very efficient way to simulate high multiplicity events exist, but including quantum interference effects is exponentially hard in many cases

The traditional way to compute high multiplicity events is inherently probabilistic and can not include many quantum interference effects



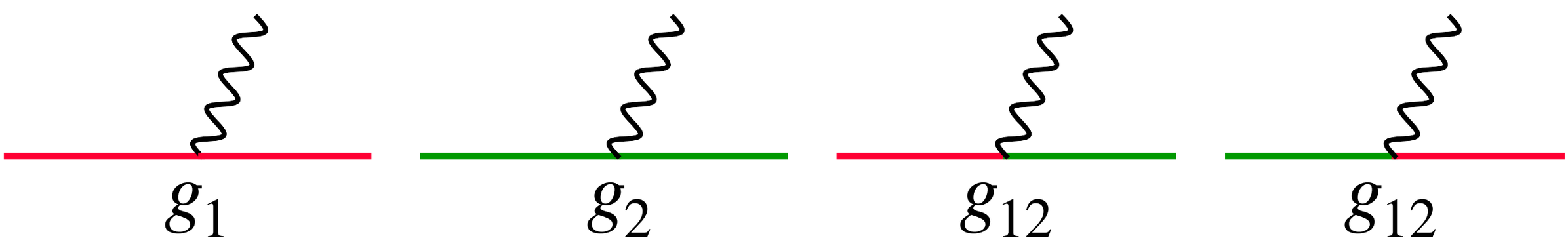
Quantum algorithms have possibility to include such quantum effects efficiently

A very simple toy model

Yukawa theory with two types of fermions and mixing between them

$$\mathcal{L} = \bar{f}_1(i\partial\!\!\!/ + m_1)f_1 + \bar{f}_2(i\partial\!\!\!/ + m_2)f_2 + (\partial_\mu\phi)^2$$
$$+ g_1\bar{f}_1 f_1\phi + g_2\bar{f}_2 f_2\phi + g_{12}\left[\bar{f}_1 f_2 + \bar{f}_2 f_1\right]\phi$$

Very simple Feynman rules

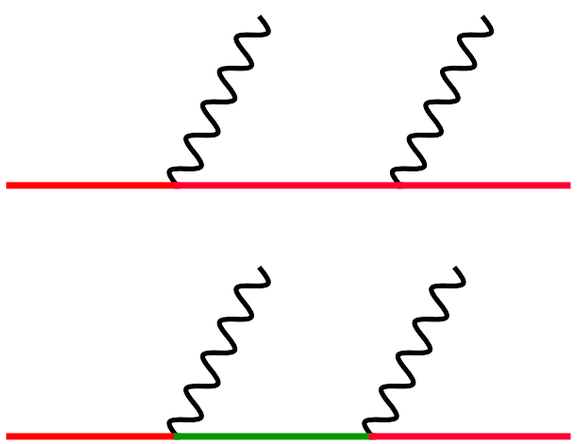


A very simple toy model

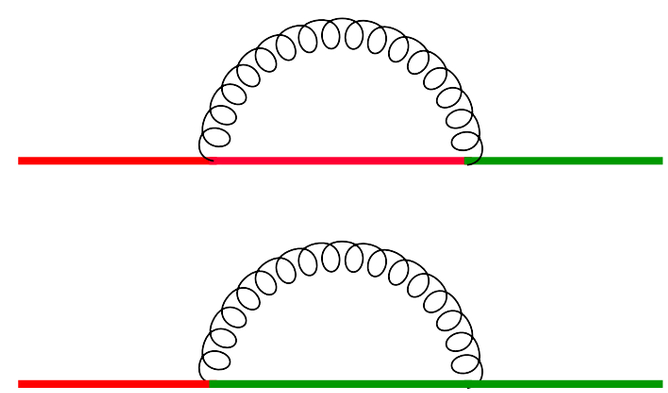
$$\mathcal{L} = \bar{f}_1(i\partial\!\!\!/ + m_1)f_1 + \bar{f}_2(i\partial\!\!\!/ + m_2)f_2 + (\partial_\mu\phi)^2 + g_1\bar{f}_1 f_1\phi + g_2\bar{f}_2 f_2\phi + g_{12}\left[\bar{f}_1 f_2 + \bar{f}_2 f_1\right]\phi$$

The mixing g_{12} gives several interesting effects

Different real emission amplitudes give rise to interference



Virtual diagrams give rise to flavor change without radiation



Need to correct both real and virtual effects

Similar to including subleading color

A 2x2 matrix can be diagonalized…

Interaction can be written in matrix notation

$$(\bar{f}_1, \bar{f}_2) \begin{pmatrix} g_1 & g_{12} \\ g_{12} & g_2 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \phi$$

This can be diagonalized as

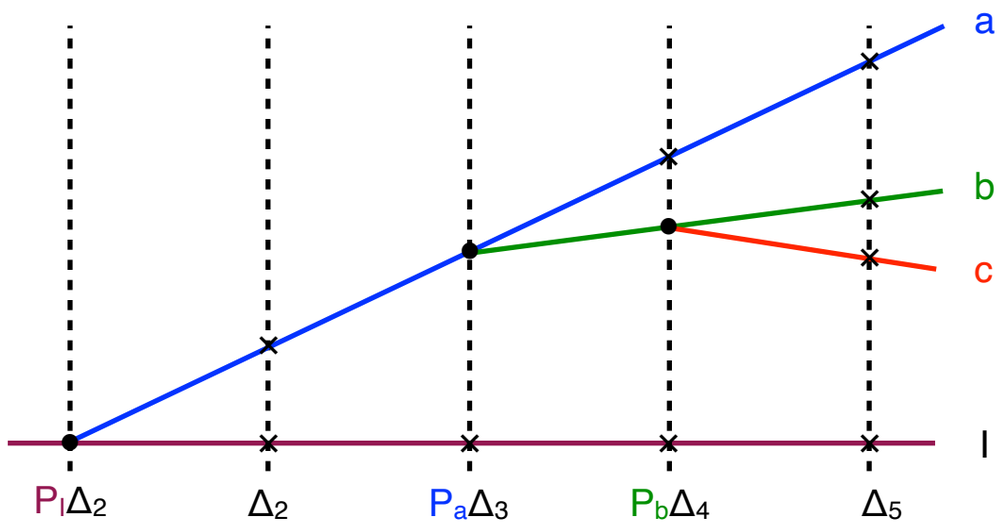
$$(\bar{f}_1, \bar{f}_2)\, U^\dagger \begin{pmatrix} g_1 & g_{12} \\ g_{12} & g_2 \end{pmatrix} U \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \phi \equiv (\bar{f}_a, \bar{f}_b) \begin{pmatrix} g_a & 0 \\ 0 & g_b \end{pmatrix} \begin{pmatrix} f_a \\ f_b \end{pmatrix} \phi$$

$$g_a = \frac{g_1 + g_2 - g'}{2}\,, \qquad g_b = \frac{g_1 + g_2 + g'}{2}\,, \qquad g' = \mathrm{sign}(g_2 - g_1)\sqrt{(g_1 - g_2)^2 + 4g_{12}^2}$$

$$U = \begin{pmatrix} \sqrt{1-u^2} & u \\ -u & \sqrt{1-u^2} \end{pmatrix}\,, \qquad u = \sqrt{\frac{(g_1 - g_2 + g')}{2g'}}$$

Thus, the theory can be transformed into a system of non-interacting fermions

This gives normal evolution



At each discrete step, need to

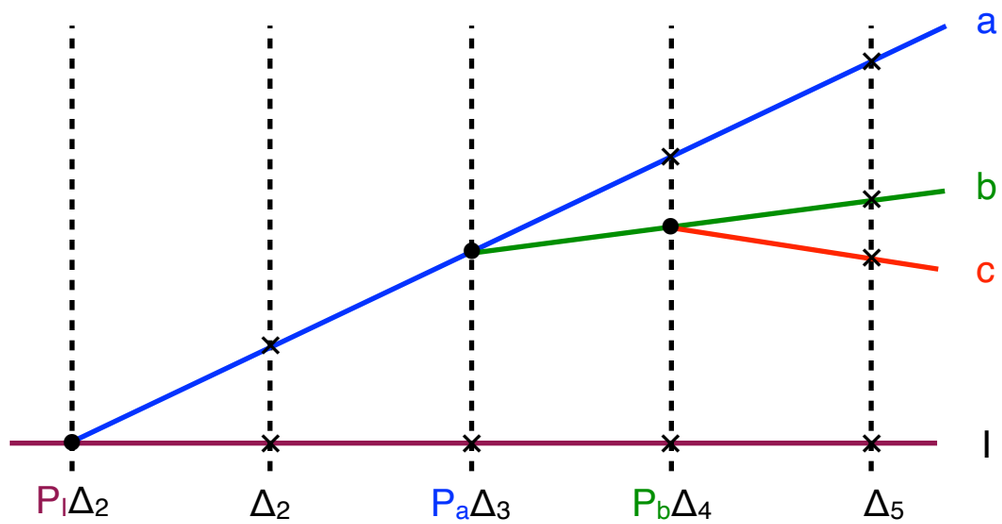
(a) Do nothing (determined by Δ_i)
(b) Emit one particle (determined by $P_{p,i}$)

Denote initial state as state in n-particle Hilbert space

```
state = to_diagonal_basis(initial_state)
for i in 1... N:
    if emission(state):
        n = choose_emitter(state)
        state = new_state(state, n)
final_state = from_diagonal_basis(state)
```

Final state is state in (n+N)-particle Hilbert space

Results in exponentially hard problem



- Δ_i only depends on n_a, n_b, but different for each i
- $P_{p,\,i}$ depends on flavor of each particle, but independent of i

There are two important facts to realize:

1. We need to rotate back to the f_1, f_2 basis in the end, so need to compute amplitudes, not probabilities
2. Need the results for all possible final state particles f_a, f_b

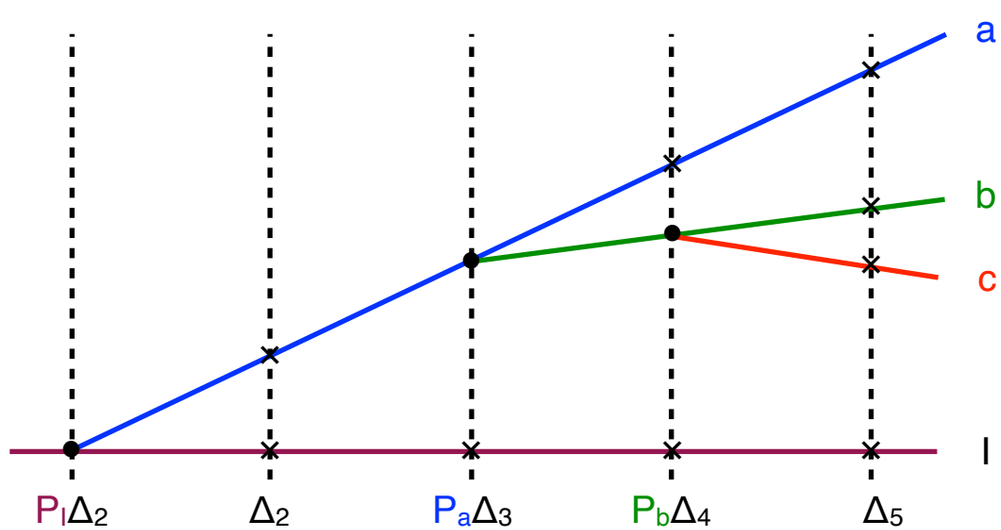
This means that for each shower history, need amplitudes for all possible flavors of fermions

This grows like 2^{n_f} for n_f fermions

A quantum computer can compute the 2^{nf} amplitudes using polynomial number of operators

- Discretize time (evolution variable) and allow emissions at each discrete value
- Only include interference effects from f_a f_b interference

Final state determined by history of emissions and types of final state particles

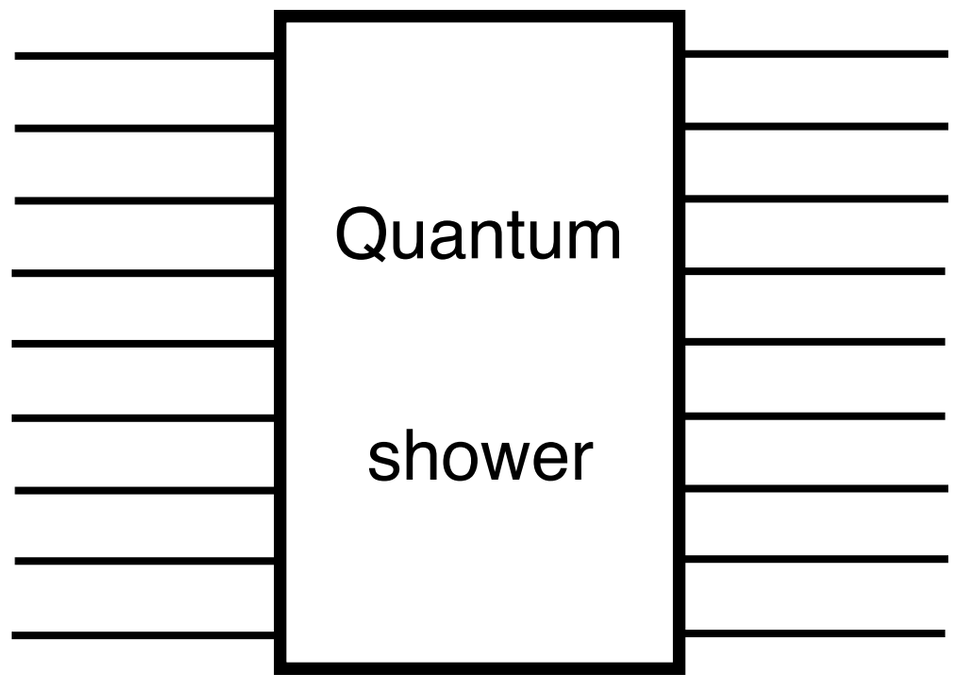


Qubit registers for history |h>, types of final state particles |p> and ancillary information

Register	Purpose	# of qubits
$\|p\rangle$	Particle state	$3(N+n_I)$
$\|h\rangle$	Emission history	$N\lceil\log_2(N+n_I)\rceil$
$\|e\rangle$	Did emission happen?	1
$\|n_\phi\rangle$	Number of bosons	$\lceil\log_2(N+n_I)\rceil$
$\|n_a\rangle$	Number of f_a	$\lceil\log_2(N+n_I)\rceil$
$\|n_b\rangle$	Number of f_b	$\lceil\log_2(N+n_I)\rceil$

A quantum computer can compute the 2^{nf} amplitudes using polynomial number of operators

Goal of algorithm is to create superposition of final states with correct relative amplitudes

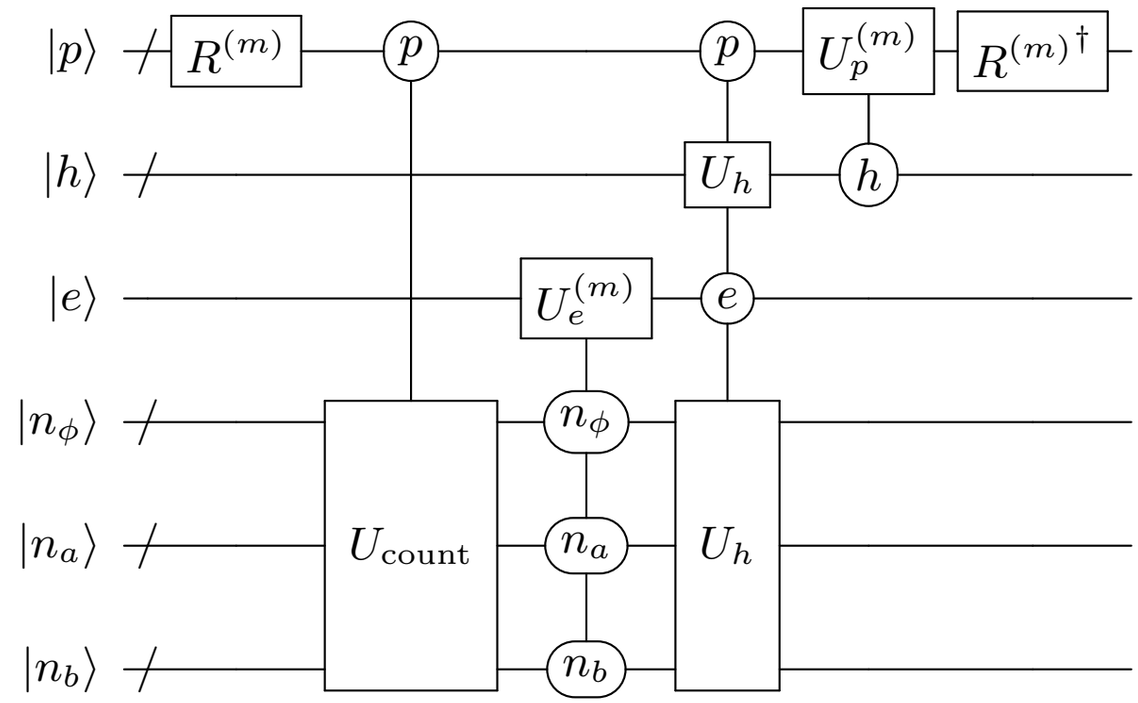


$$|000\ldots0\rangle \rightarrow A_1\,|\Psi_1\rangle + \ldots A_n\,|\Psi_n\rangle$$

Repeated measurements of the final state selects states with probability $|A_i|^2$ ⇒ can be used as true event generator

A quantum computer can compute the 2^{nf} amplitudes using polynomial number of operators

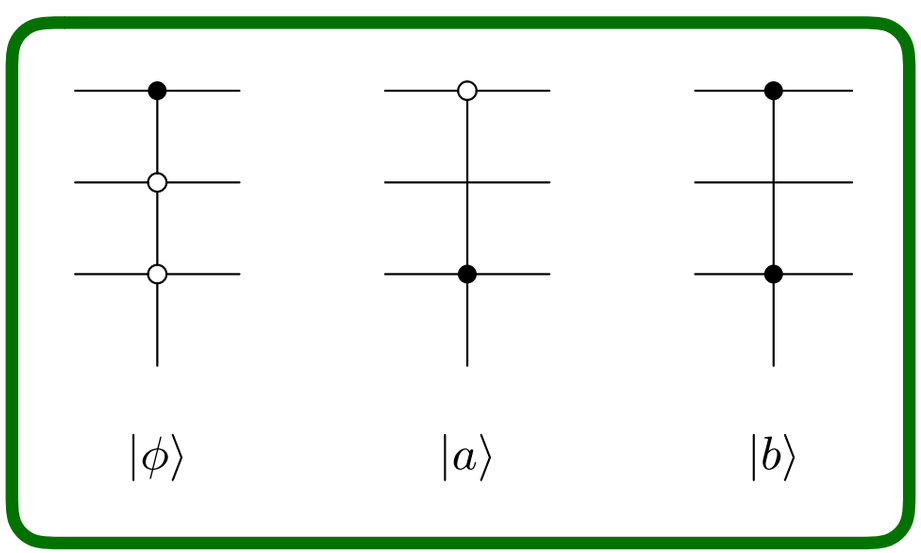
At each discreet time interval, algorithm rotates from f_1, f_2 basis to f_a, f_b basis, performs shower in 4 separate steps, and rotates back to f_1, f_2 basis



$|n_i\rangle$, $|h\rangle$: Integer registers

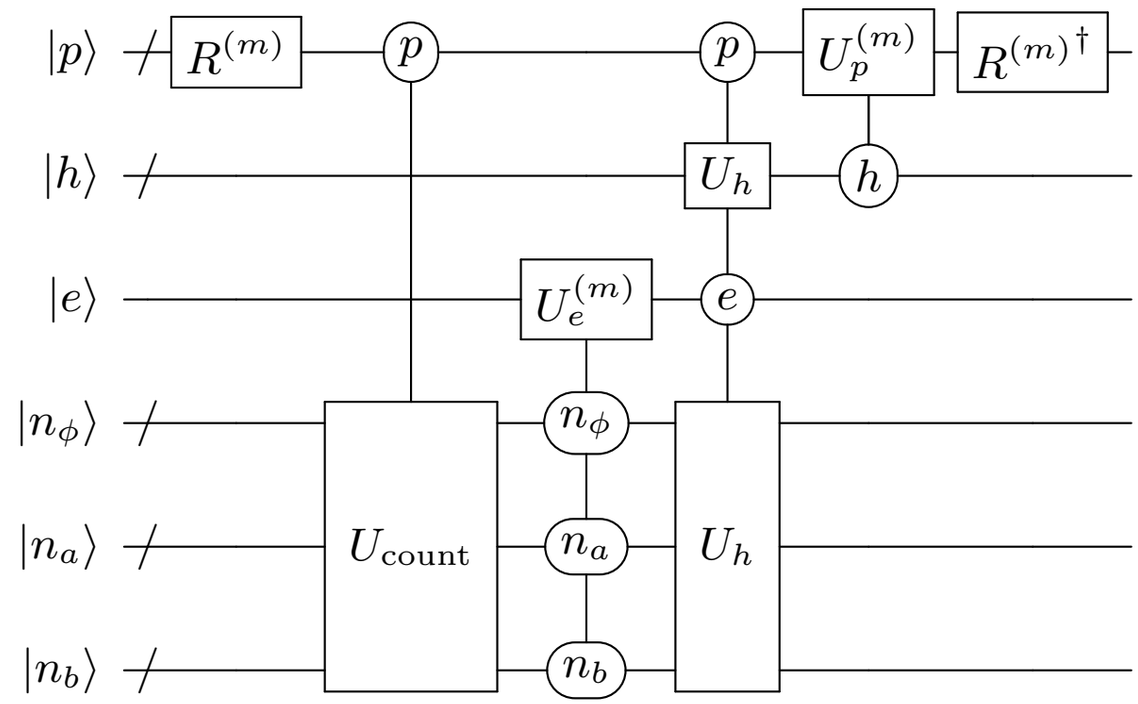
$|e\rangle$: Boolean value

$$|p\rangle_i = \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} = \begin{pmatrix} 0 \\ \phi \\ - \\ - \\ f_1/f_a \\ f_2/f_b \\ \bar{f}_1/\bar{f}_a \\ \bar{f}_2/\bar{f}_b \end{pmatrix}$$



A quantum computer can compute the 2^{n_f} amplitudes using polynomial number of operators

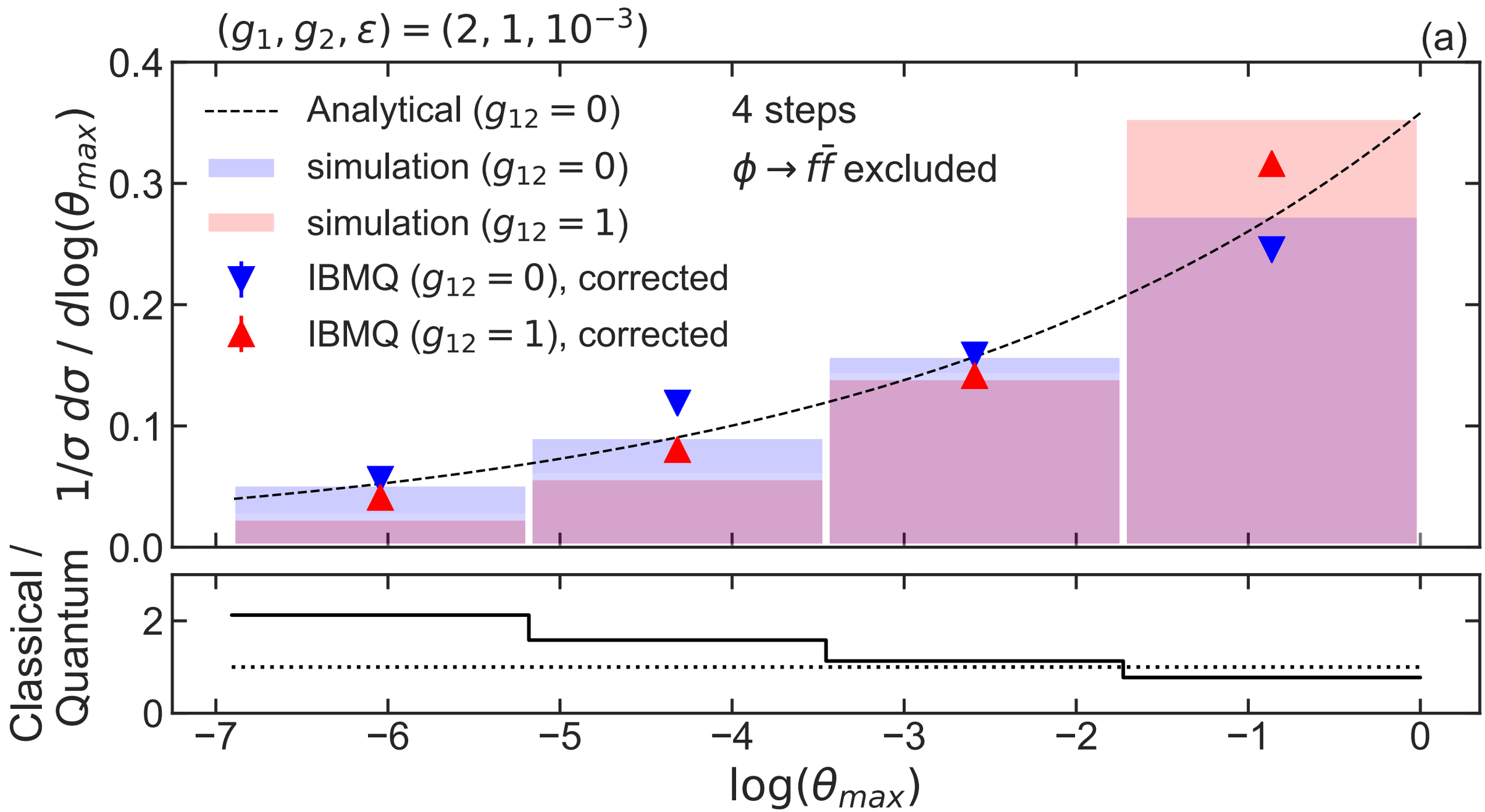
At each discreet time interval, algorithm rotates from f_1, f_2 basis to f_a, f_b basis, performs shower in 4 separate steps, and rotates back to f_1, f_2 basis



Operation	Scaling
count particles U_{count}	$N \ln(n_f)$
decide emission U_e	$N\, n_f \ln(n_f)$
create history U_h	$N\, n_f^2 \ln(n_f)$
adjust particles U_p	$N\, n_f \ln(n_f)$

classical algorithms scales as

$$N\, 2^{n_f/2}$$


$(g_1, g_2, \varepsilon) = (2, 1, 10^{-3})$
(a)
Analytical ($g_{12} = 0$)
simulation ($g_{12} = 0$)
simulation ($g_{12} = 1$)
IBMQ ($g_{12} = 0$), corrected
IBMQ ($g_{12} = 1$), corrected
4 steps
$\phi \to f\bar{f}$ excluded
$1/\sigma\, d\sigma / d\log(\theta_{max})$
Classical / Quantum
$\log(\theta_{max})$

There are many things that needs to happen before this becomes truly useful

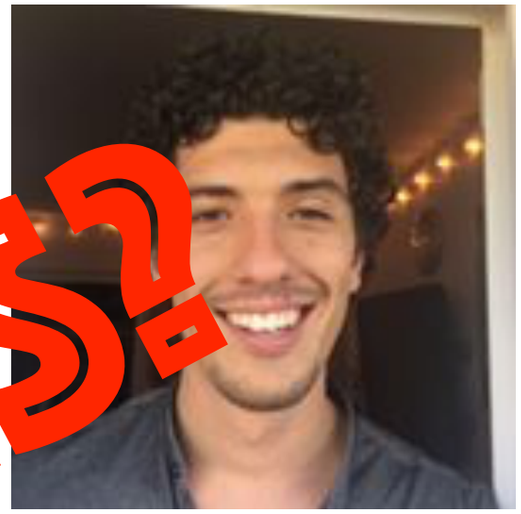
1. Apply to quantum interference effects of standard model
2. Reduce the circuit depth and required qubits
3. Find ways to make code more robust against noise
4. ……..

But our proof of principle that quantum interference effects in parton showers can be included using quantum algorithms is important first step

There are many other interesting questions in quantum computing that we (and many others) are working on

1. Dynamical simulation of quantum field theories on lattices
2. Find better ways of sampling from given distributions
3. Ways to correct for readout and gate noise
4. Efficient ways to prepare complicated states
5. ……..



QUESTIONS?