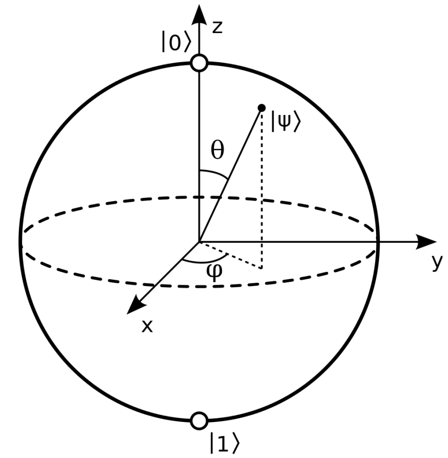
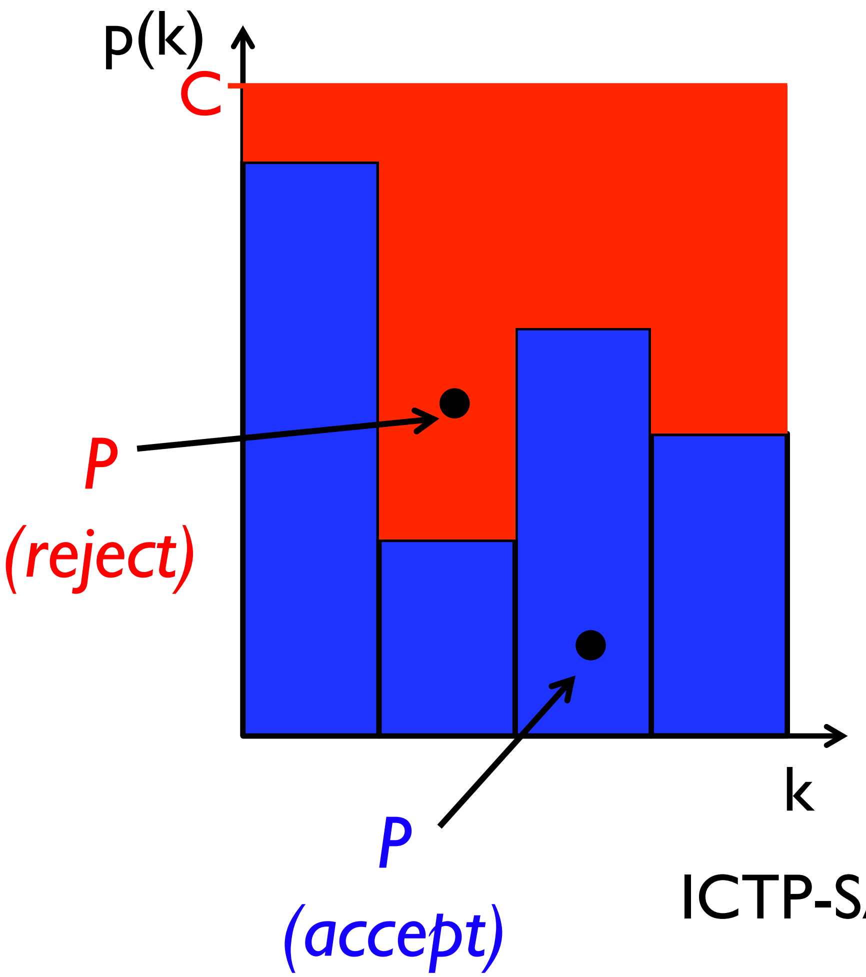
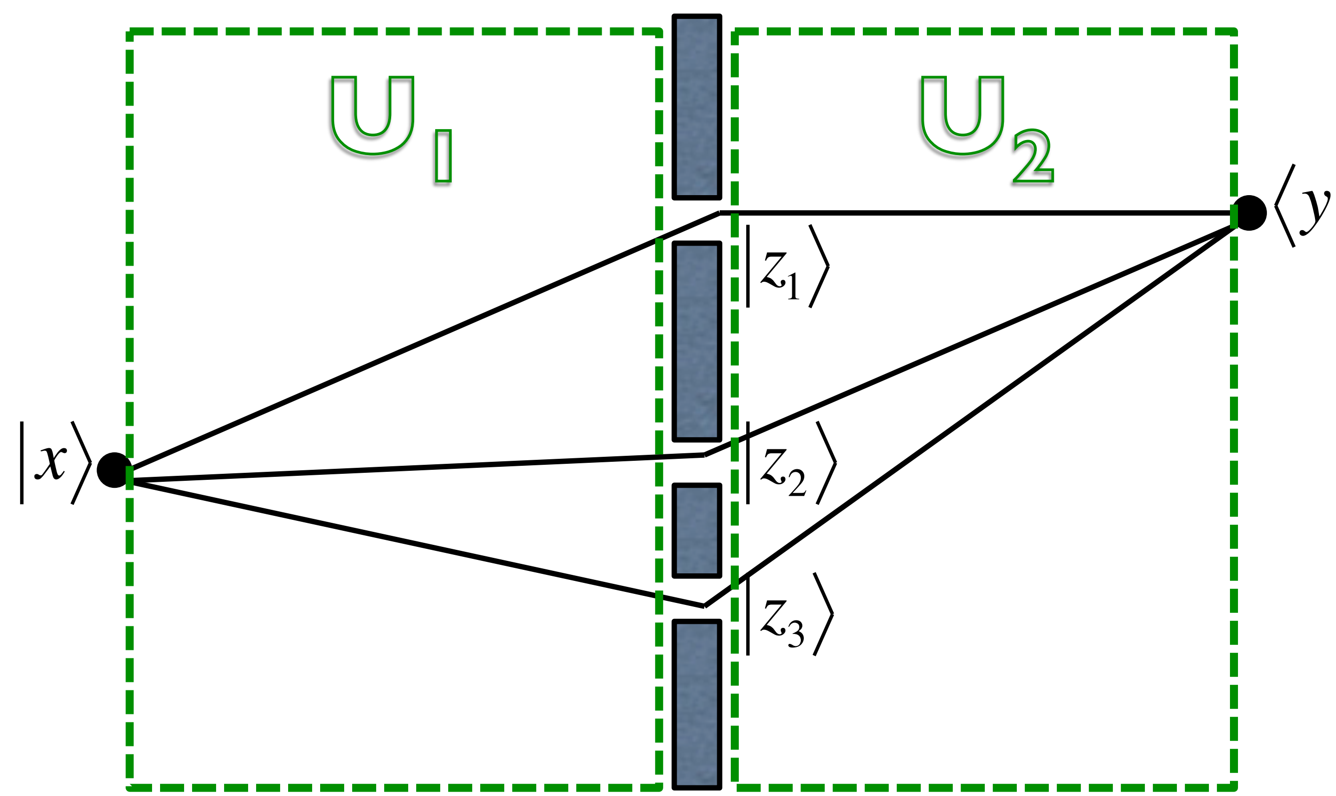



# Introduction to quantum computation and simulability

Daniel J. Brod (UFF)
Leandro Aolita (UFRJ/ICTP-SAIFR)
<u>Ernesto F. Galvão (UFF)</u>

INSTITUTO DE FÍSICA
Universidade Federal Fluminense




ICTP-SAIFR – IFT/UNESP , October 15th-19th, 2018

# Lecture 12 : Other approaches to simulation

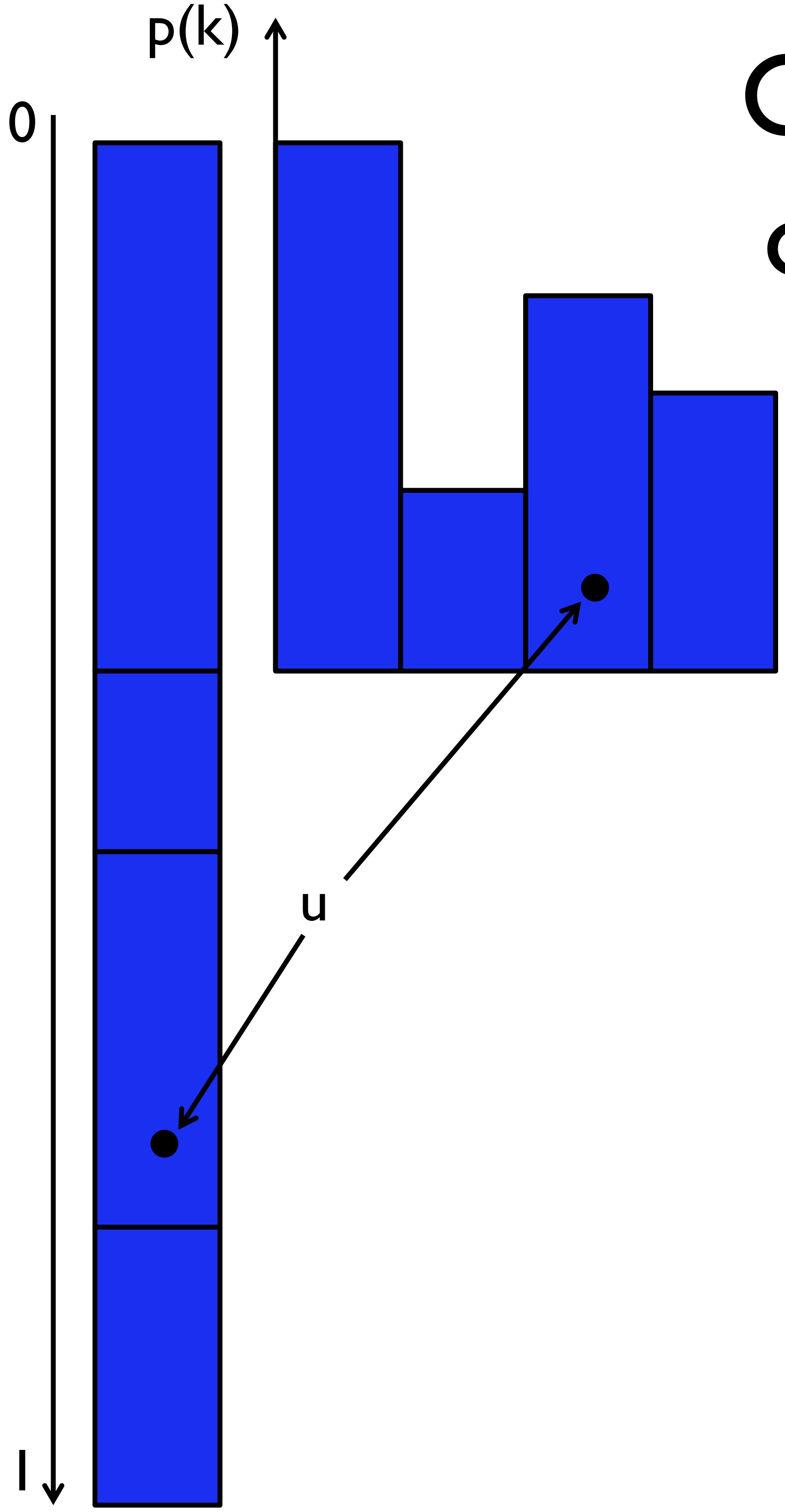
Outline:

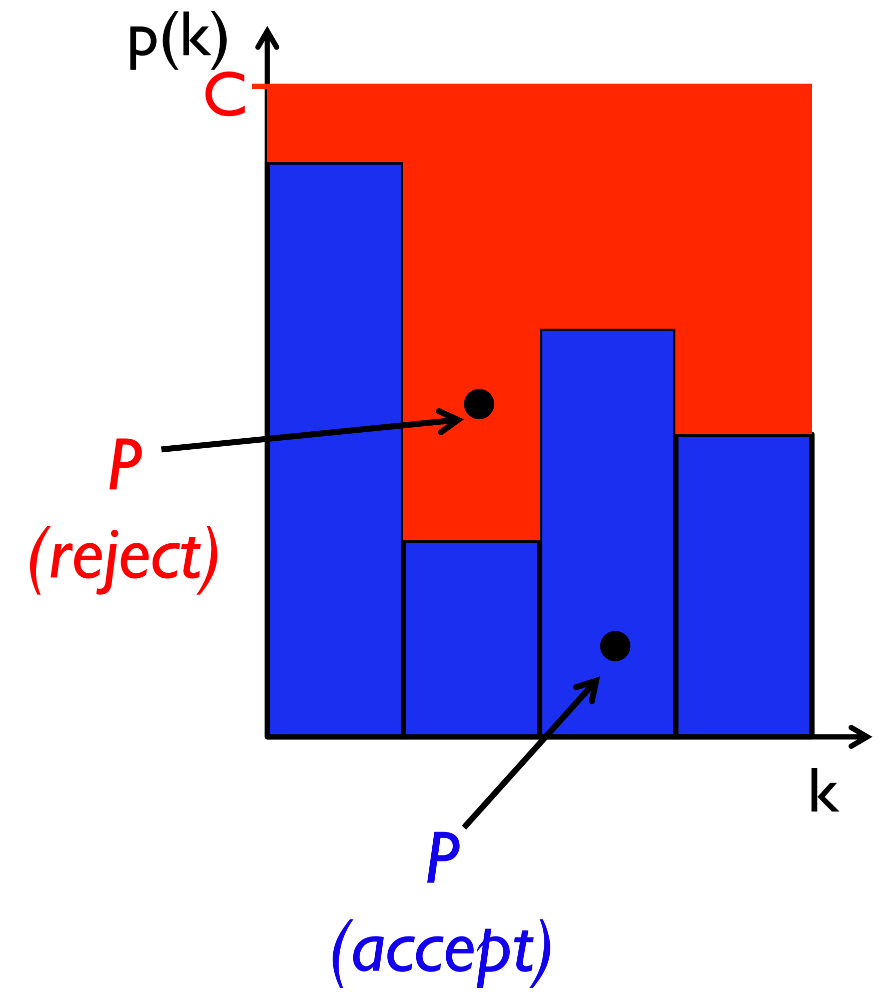
- 3 approaches to simulation of Boson Sampling
  - brute force
  - brute force with little memory
  - rejection sampling

- Simulation algorithms for general quantum circuits
  - Schrodinger scheme
  - Schemes based on Feynman's path integrals

- For slides and links to related material, see

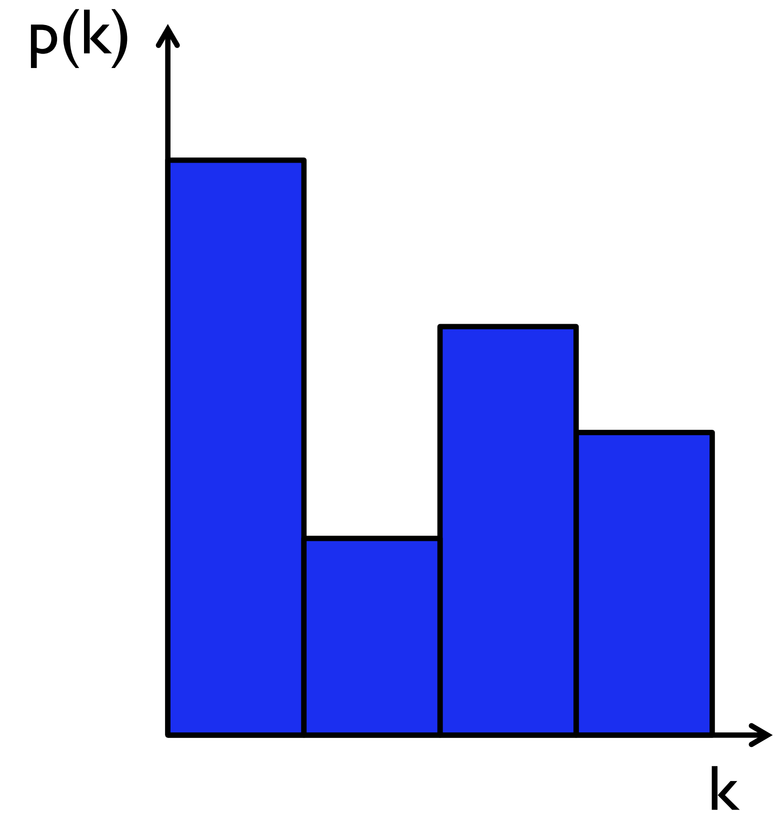
# Classical simulation of Boson Sampling



# Classical simulation of Boson Sampling

- Assume $n$ photons interfering in $m=n^2$ modes
- Boson Sampling distribution p(k) over outputs k:
  - number of possible outputs: $\binom{n^2}{n} \boxed{\propto \exp(n)}$
  - each p(k) given by $|\text{permanent}(U_k)|^2$ of $n$x$n$ matrix $U_k$
    (computationally demanding)




**How can we simulate Boson Sampling on a classical computer?**

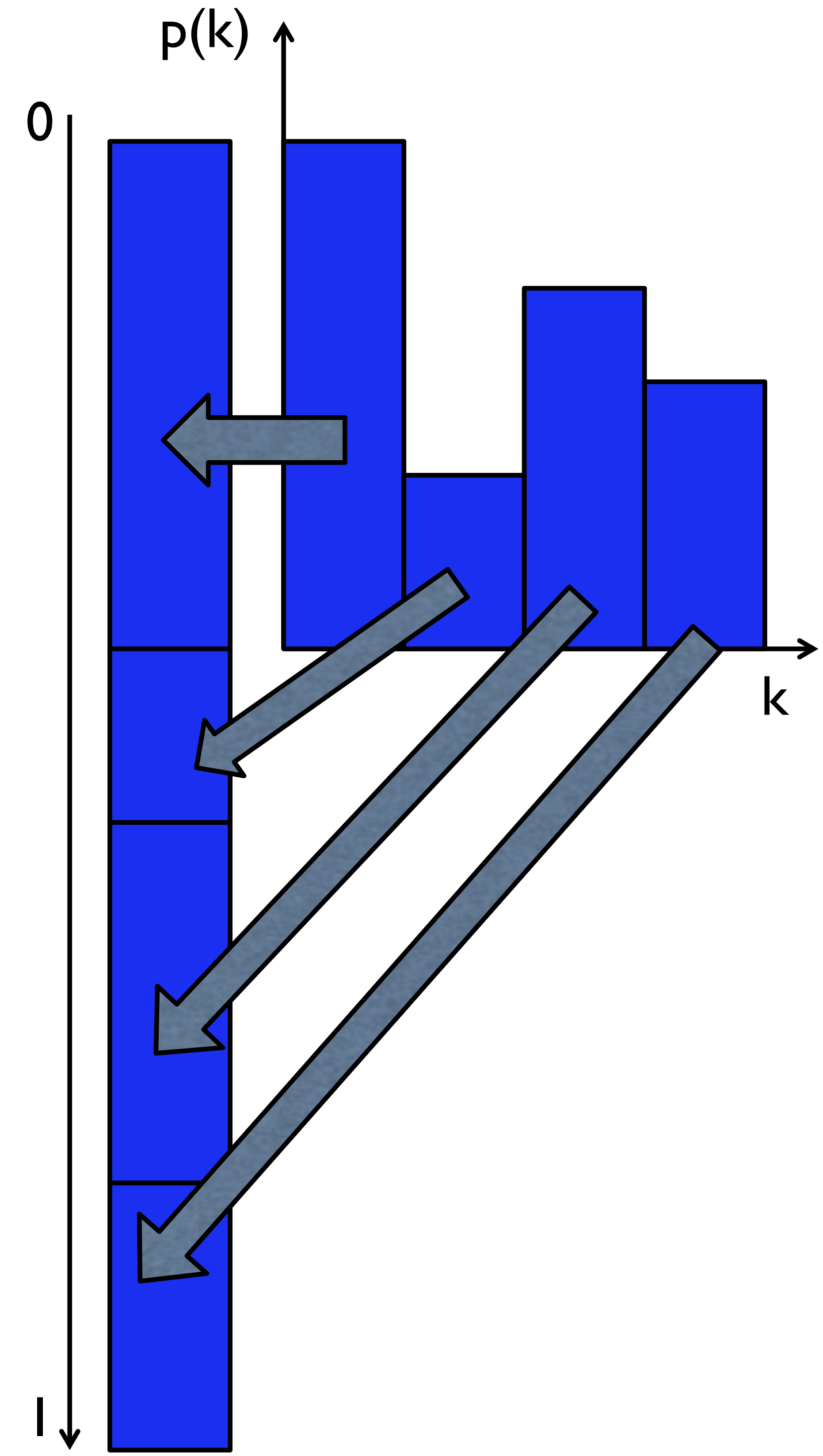
## Algorithm A: brute force

- Calculate and store each p(k), then sample from that distribution
- Memory = exp(n), time = exp(n).

# Classical simulation of Boson Sampling

**Algorithm B: brute force with small memory**

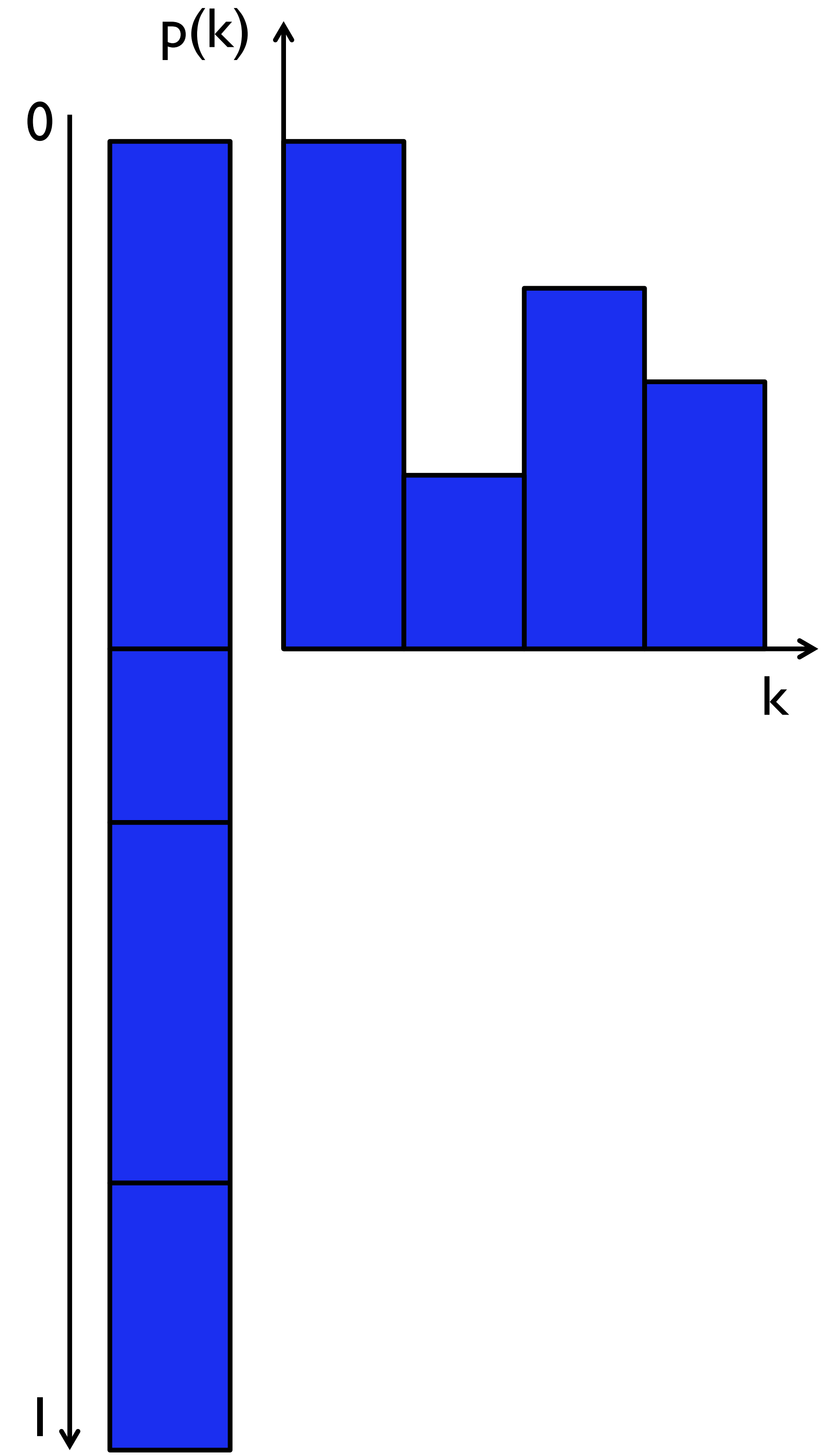
- Exact sample = uniform sample from blue area



# Classical simulation of Boson Sampling

**Algorithm B: brute force with small memory**

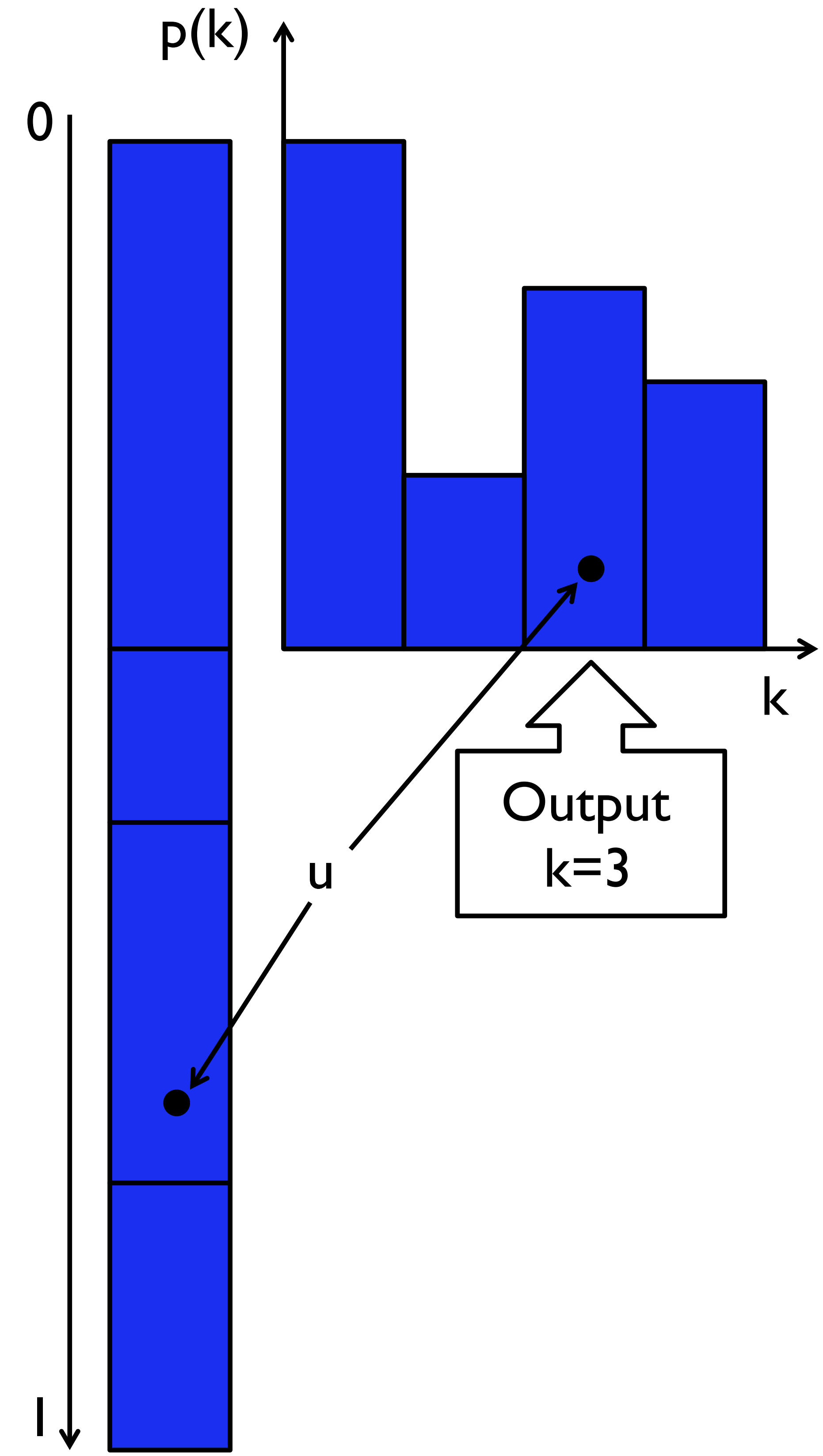
- Exact sample = uniform sample from blue area



# Classical simulation of Boson Sampling

## Algorithm B: brute force with small memory

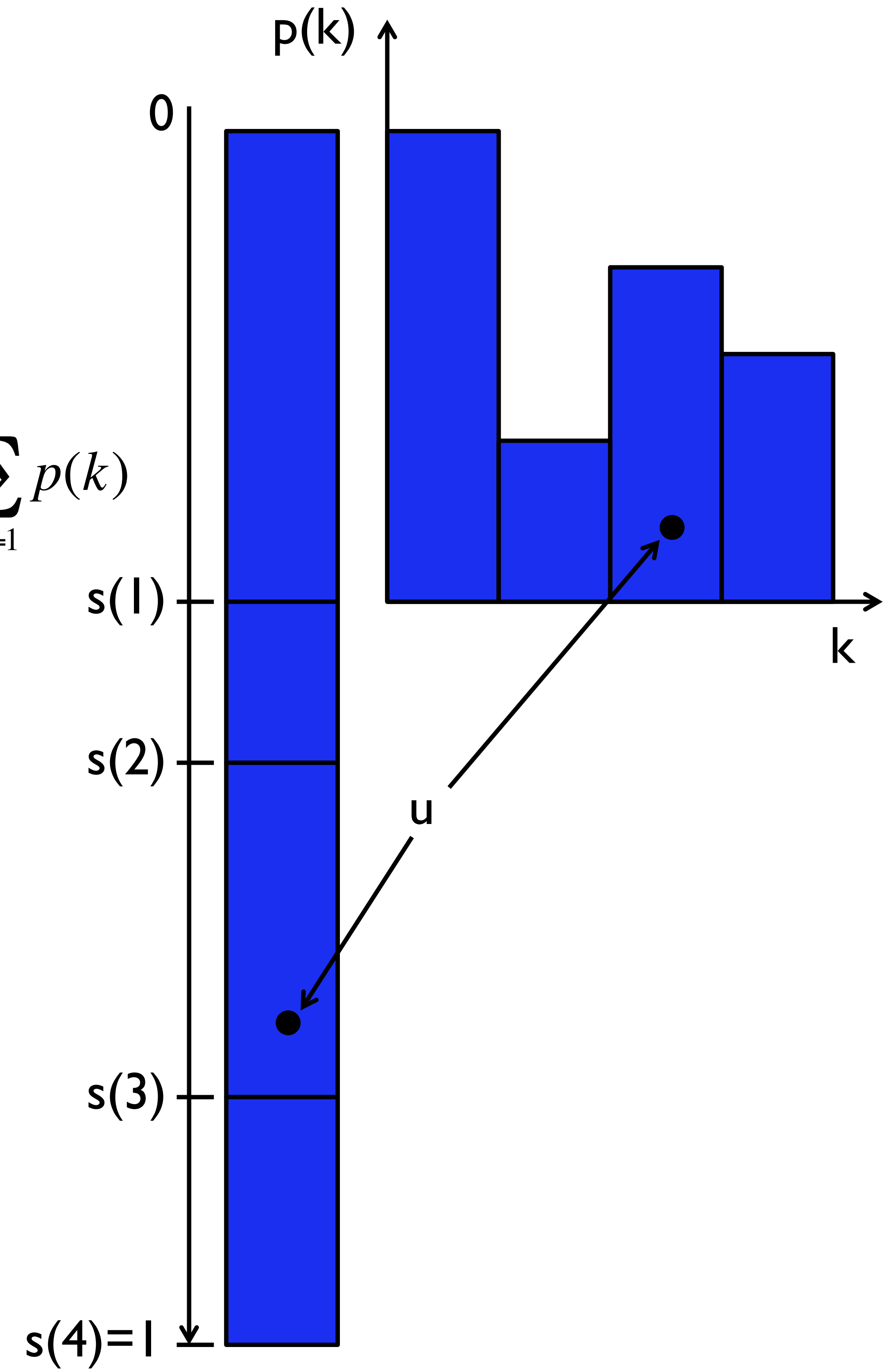
- Exact sample = uniform sample from blue area
- Equivalent to uniformly sampling u in (0,1)…
  … and outputting the k which contains u



# Classical simulation of Boson Sampling

**Algorithm B: brute force with small memory**

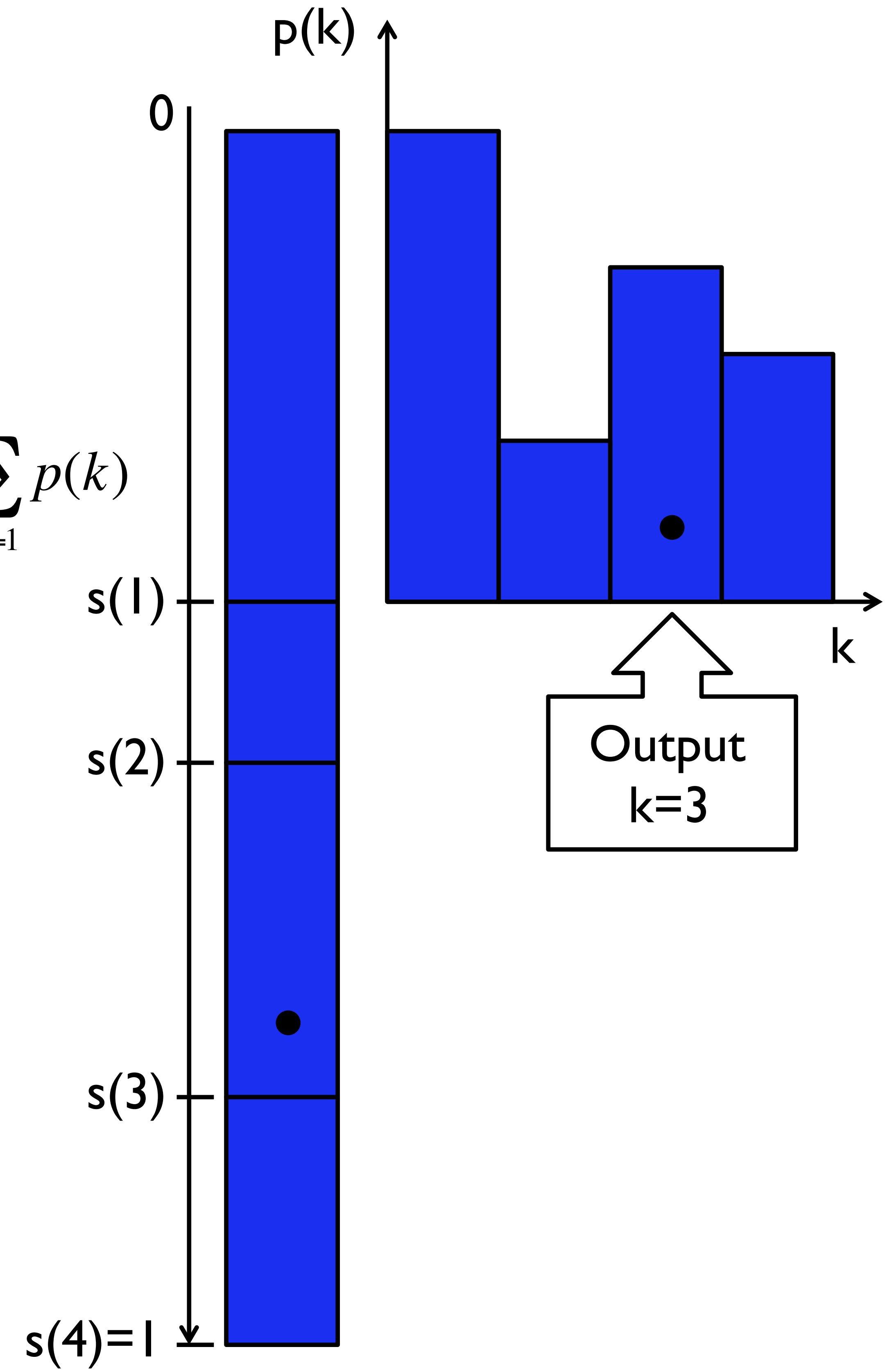
- Exact sample = uniform sample from blue area
- Equivalent to uniformly sampling u in (0,1)…
  … and outputting the k which contains u
- Sufficient to compute&store cumulative sum $s(i) = \sum_{k=1}^{i} p(k)$
  until $s(i) \geq u$



# Classical simulation of Boson Sampling

**Algorithm B: brute force with small memory**

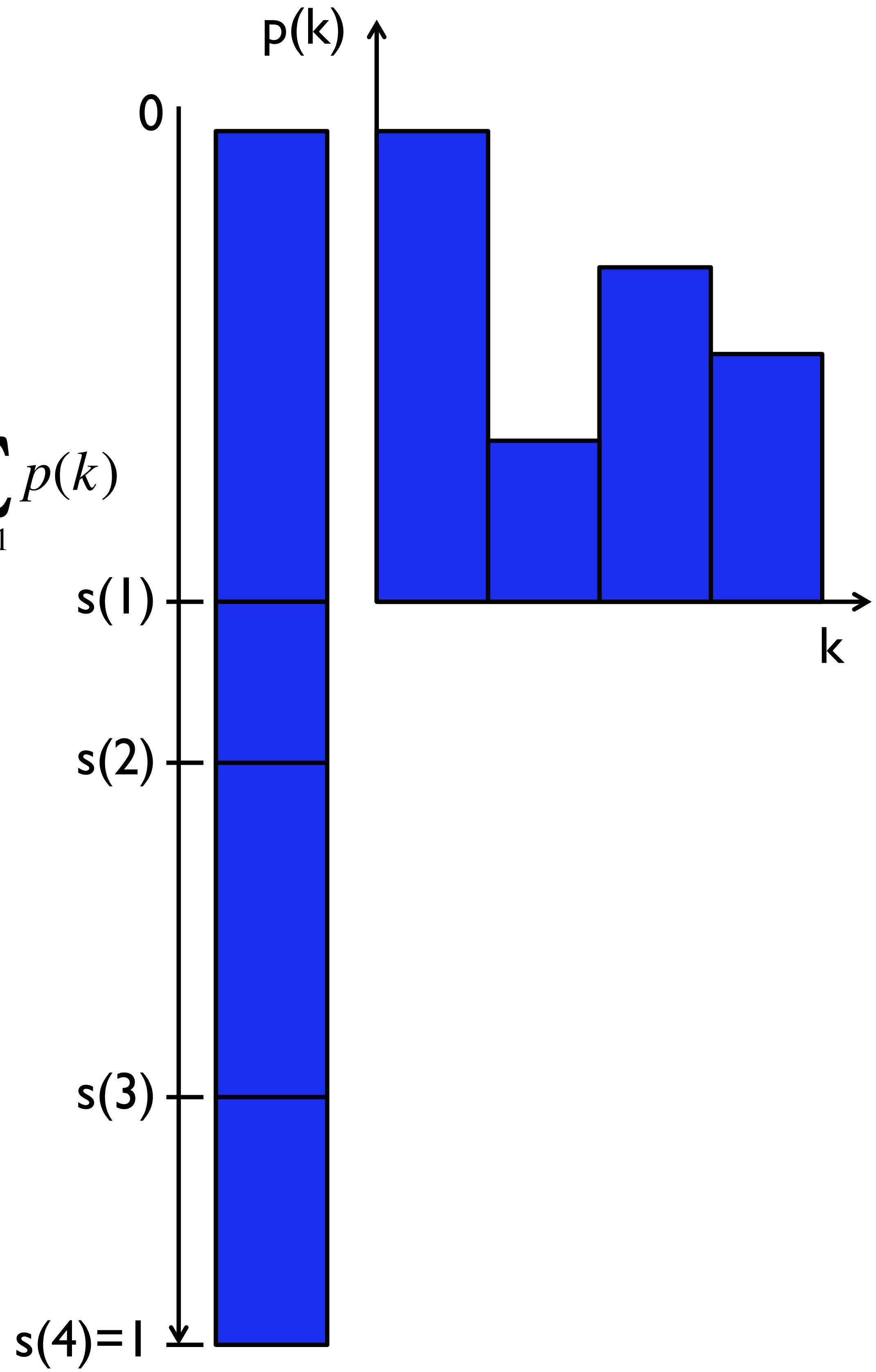
- Exact sample = uniform sample from blue area
- Equivalent to uniformly sampling u in (0,1)…
  … and outputting the k which contains u
- Sufficient to compute&store cumulative sum $s(i) = \sum_{k=1}^{i} p(k)$
  until $s(i) \geq u$



# Classical simulation of Boson Sampling

**Algorithm B: brute force with small memory**

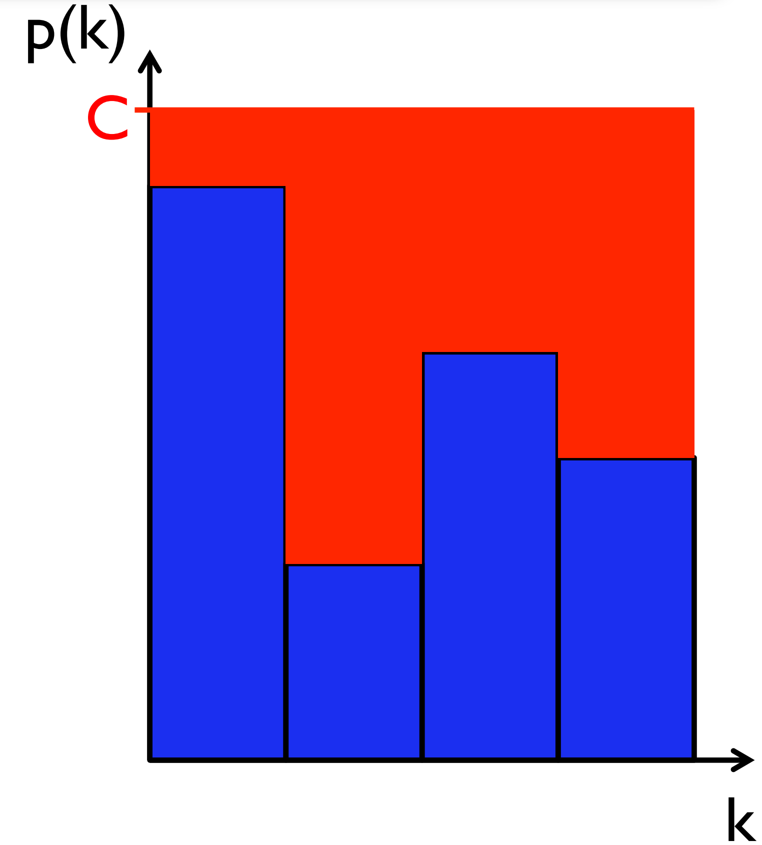
- Exact sample = uniform sample from blue area
- Equivalent to uniformly sampling u in (0,1)…
  … and outputting the k which contains u
- Sufficient to compute&store cumulative sum $s(i) = \sum_{k=1}^{i} p(k)$
  until $s(i) \geq u$
- Easy to sample *N* events, in worst case we need to compute all permanents once.
- Time = exp(n), memory = poly(n)



# Classical simulation of Boson Sampling

**Algorithm C: rejection sampling**

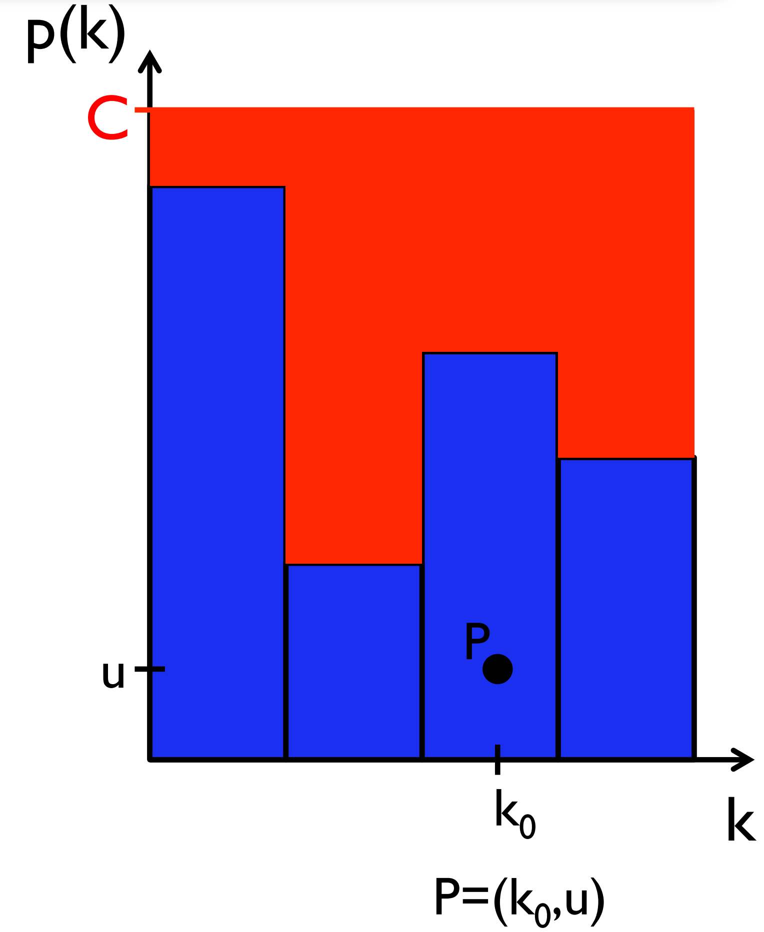
- Rejection sampling algorithm:
  - Choose constant $C \geq p(k) \quad \forall k$
  - Pick point *P* uniformly in red/blue box:
    - pick $k_0$ uniformly randomly in $\{1, 2, ..., k_{\max}\}$
    - pick u uniformly randomly in $(0, C)$



# Classical simulation of Boson Sampling

## Algorithm C: rejection sampling

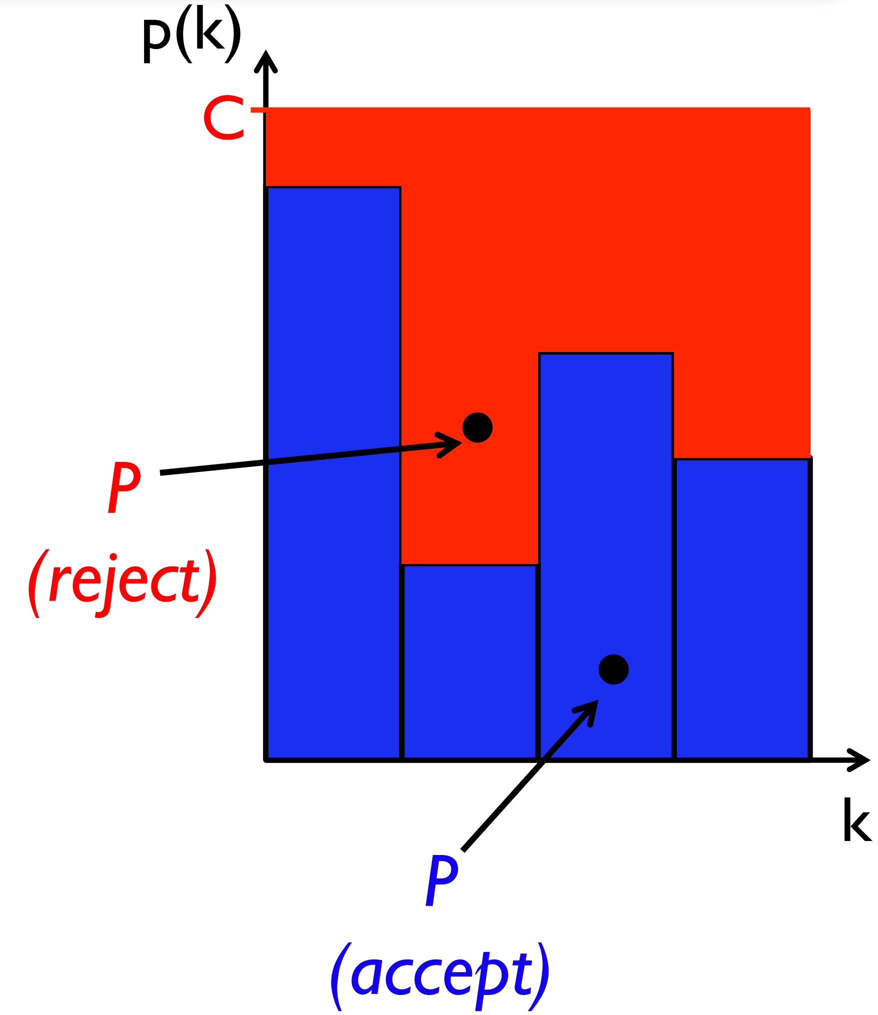
- Rejection sampling algorithm:
  - Choose constant $C \geq p(k) \quad \forall k$
  - Pick point *P* uniformly in red/blue box:
    - pick $k_0$ uniformly randomly in $\{1, 2, \ldots, k_{\max}\}$
    - pick u uniformly randomly in $(0, C)$



# Classical simulation of Boson Sampling

## Algorithm C: rejection sampling

- Rejection sampling algorithm:
  - Choose constant $C \geq p(k) \quad \forall k$
  - Pick point *P* uniformly in red/blue box:
    - pick $k_0$ uniformly randomly in $\{1,2,\ldots,k_{\max}\}$
    - pick u uniformly randomly in $(0,C)$
  - Calculate blue/red border: $p(k_0)=|\mathrm{per}(U_{k0})|^2$
    - If *P* in blue – accept and output $k_0$
    - If *P* in red – reject and try again
- Sampling is **exact** if $C \geq p(k) \quad \forall k$




- Possible to prove high-probability upper bound C for p(k), valid for uniform ensemble of unitaries (*Scott Aaronson, private communication*)
- Memory = poly(n), time = exp(n)*$\boxed{\text{m}}$ benign overhead with m
- Total variation distance error < $1/(k_{max}*C)$

# Simulation of general quantum circuits

2 general approaches to simulation of general quantum circuits:

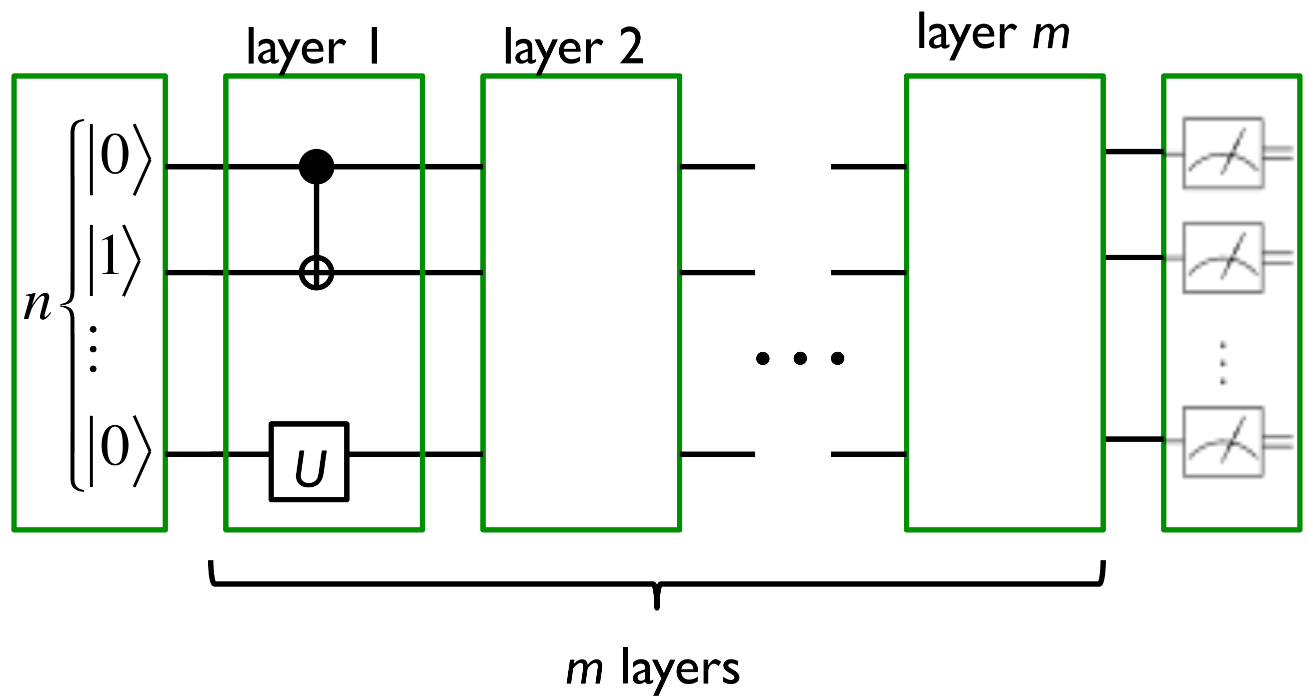
- Brute-force calculation à la Schrodinger

- Calculation with polynomial memory – à la Feynman

# Schrodinger simulation: exp(n) time, exp(n) space

This is the approach most students of QM would take. Setting:

- *n* qubits
- depth *m* (= number of temporal layers of gates)



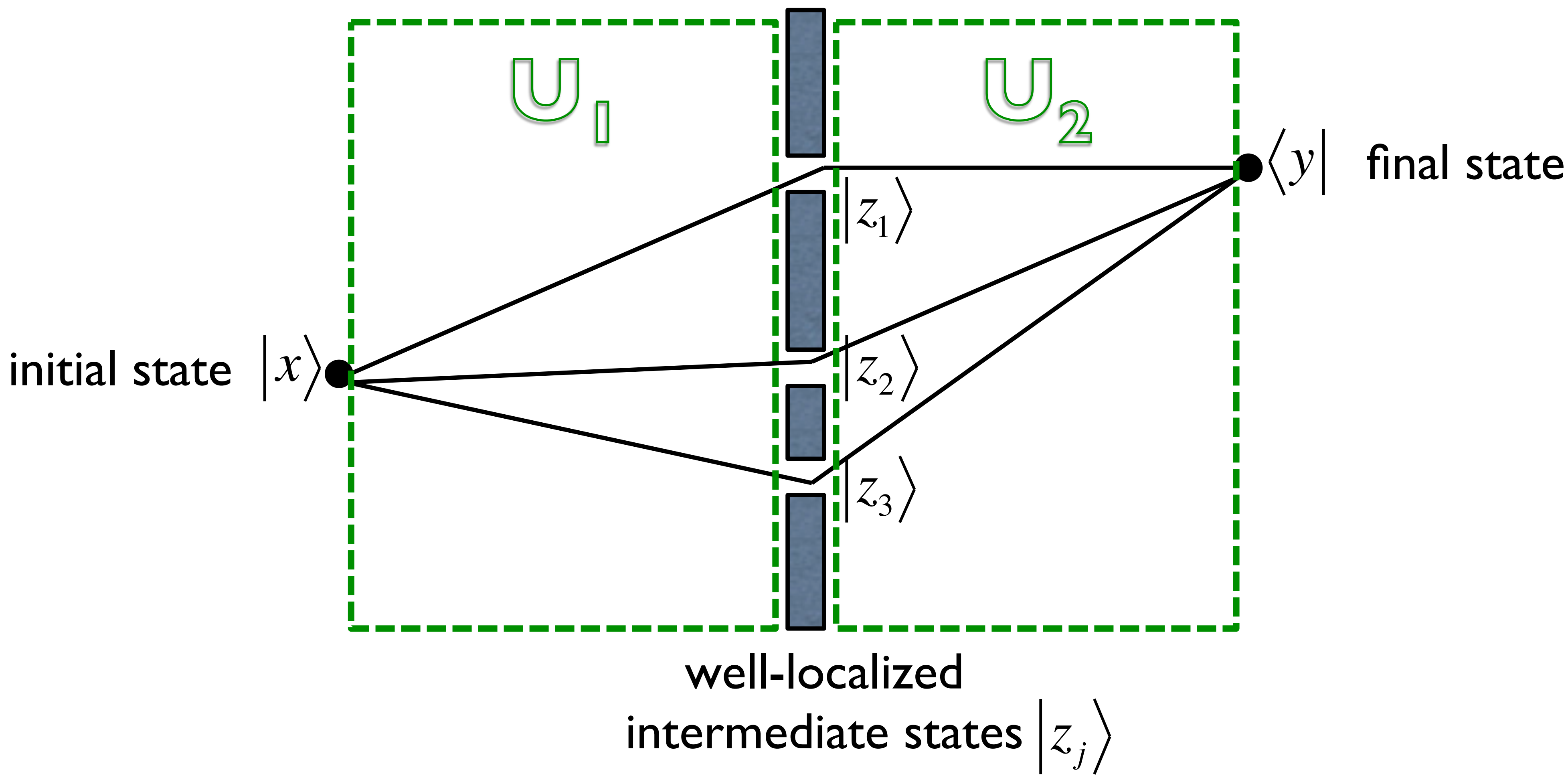

- Simulation:

1. Initialize input state;
2. Calculate state after first layer of one- and two-qubit gates;
3. Repeat step 2 above until we get the final state;
4. Directly obtain the amplitude corresponding to the final states of interest.

- Complexity:
  - *m2^n* time
  - *2^n* space (to store wavefunction amplitudes)

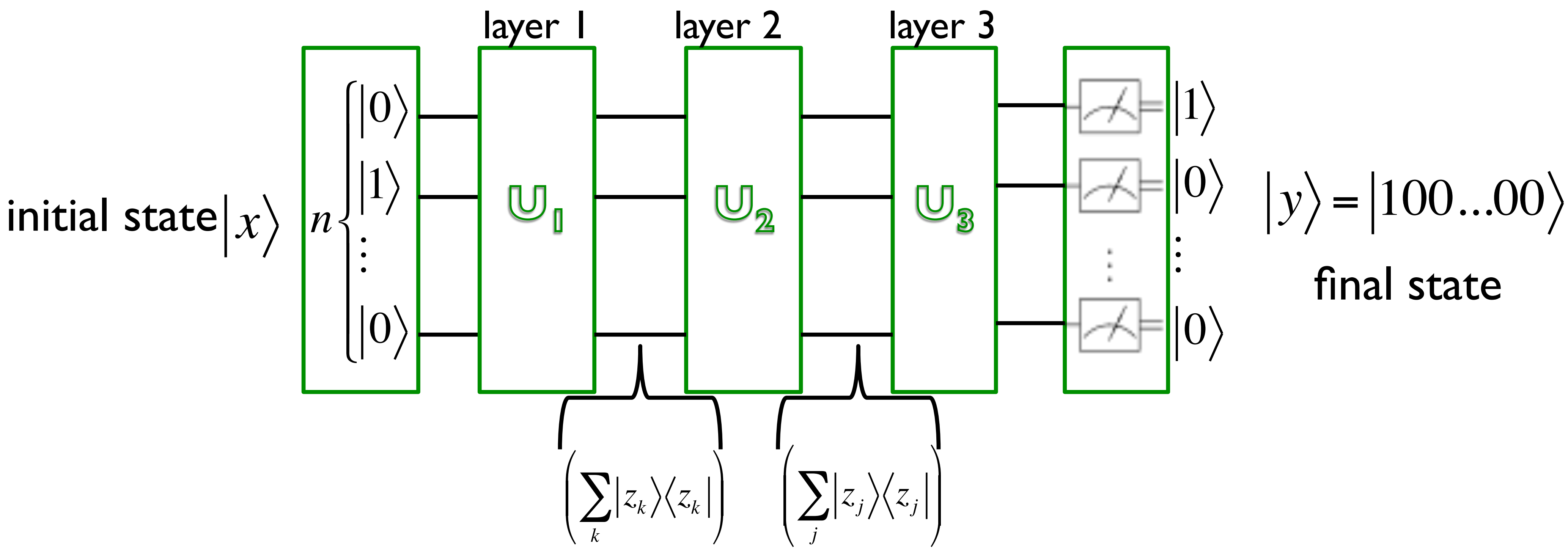
# Another approach: Feynman's path integral

- Let us look at how we can compute amplitudes using Feynman's path integrals. : calcula
- **Goal**: calculate $\langle y|U_2 U_1|x\rangle$




$$\langle y|U_2 U_1|x\rangle = \langle y|U_2 \underbrace{\left(\sum_{j=1}^{3} |z_j\rangle\langle z_j|\right)}_{=1} U_1|x\rangle = \underbrace{\sum_{j=1}^{3} \langle y|U_2|z_j\rangle\langle z_j|U_1|x\rangle}_{\text{sum over path amplitudes}}$$

# Simulation using Feynman's sum over path amplitudes




$$\langle y|U_3U_2U_1|x\rangle = \langle y|U_3\left(\sum_j |z_j\rangle\langle z_j|\right)U_2\left(\sum_k |z_k\rangle\langle z_k|\right)U_1|x\rangle = \sum_{j,k}\langle y|U_3|z_j\rangle\langle z_j|U_2|z_k\rangle\langle z_k|U_1|x\rangle$$

If we want the amplitude that the top qubit's measurement be 1:

$$\langle 1anything|U_3U_2U_1|x\rangle = \sum_w \sum_{j,k}\langle 1w|U_3|z_j\rangle\langle z_j|U_2|z_k\rangle\langle z_k|U_1|x\rangle$$

Complexity for m unitary layers:

- *exp(n)* time
- *poly(n,m)* space ( not **exponential** like the Schrodinger scheme)

# Refinements and applications

- Simulating circuits with:
  - *n* qubits
  - *m* gates
  - depth *d*

- Aaronson and Chen (2016) algorithms: [arXiv:1612.05903]

1. poly(n,m) space, m^O(n) time
2. poly(n,m) space, d^O(n) time
3. “Smooth tradeoff” with Schrodinger’s scheme:
   - Halve memory use in S. scheme => multiply time use by *d*

- Application (together with other tricks) [Pednault et al., arXiv:1710.05867]

  - 7x7=49 2D grid, random circuit, depth 27
  - 2 days of IBM Vulcan IBM Blue Gene/Q supercomputer (Lawrence Livermore Labs)
  - 4.5 TB memory use, computation of 2^38 amplitudes
  - related paper simulates 7x8=56 qubit circuit of depth 27 [Boixo et al., arXiv:1712.05903]

- Other schemes:
  - contracting tensor networks (Markov, Shi 2008), see Leandro Aolita’s lectures

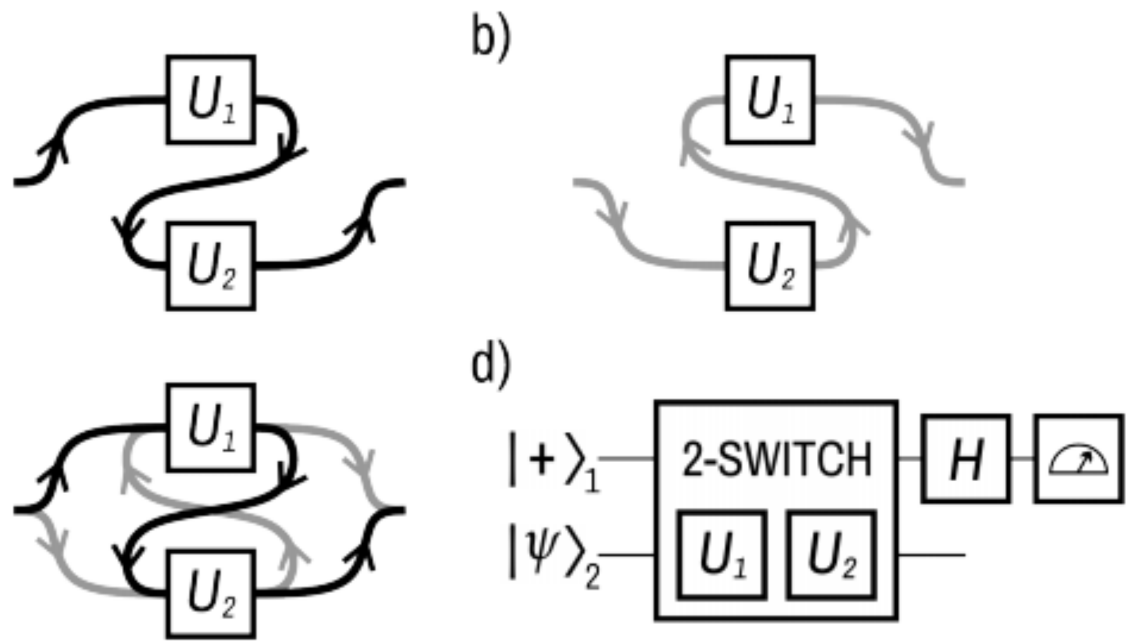
Time ordering in quantum computation:

- superposition of causal orders

- simulated closed time-like curves

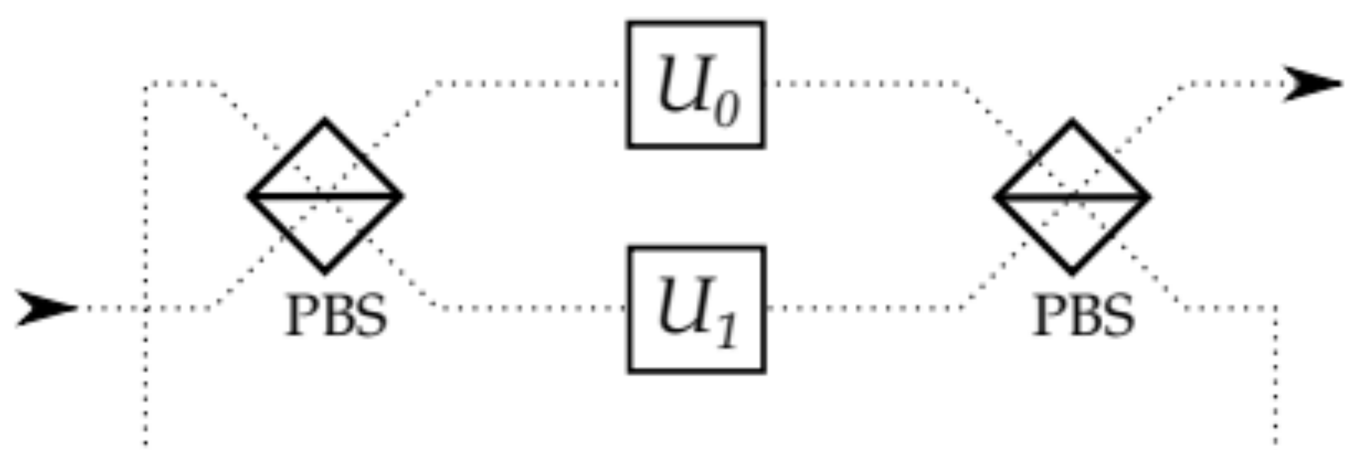
# Computational resource: superposition of causal orders

- It's possible to imagine superposing different orders of operations:




Procopio et al., arXiv:1412.4006

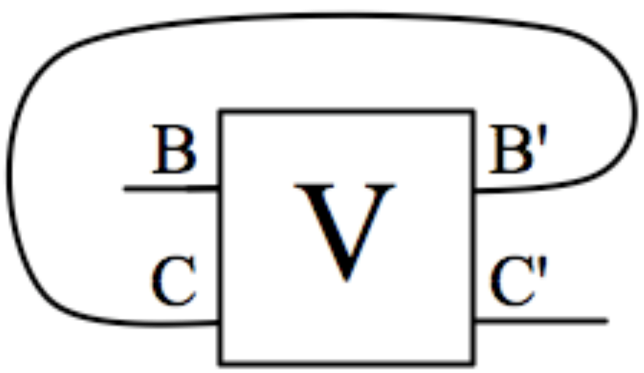
- This can be achieved using an interferometer (but not a circuit):



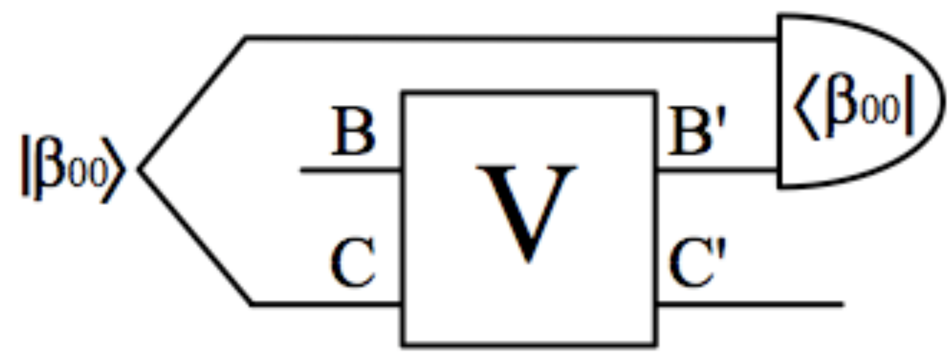

- Based on theoretical work by Chiribella (2012).

# PCTCs: a model based on teleportation and post-selection

- Bennett and Schumacher, unpublished (2002) – see seminar http://bit.ly/crs8Lb
- Rediscovered independently by Svetlichny (2009) - arXiv:0902.4898v1
  - Related work on black holes by Horowitz/Maldacena (2004), Preskill/Gottesman (2004)




CTC




$$|\beta_{00}\rangle \equiv \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

Simulation using teleportation and post-selection: B'=C

- We post-select projections onto $|\beta_{00}\rangle$
  - Postselection successful: state B' is teleported back in time (state C = state B')
  - Simulation works only when post-selection happens -> finite probability of success.