FROM QUARKS AND GLUONS TO NUCLEAR FORCES AND STRUCTURE

Lecture 1: Introduction to the Path Integral Formalism and some numerical preliminaries

Thomas Luu

JÜLICH FORSCHUNGSZENTRUM

UNIVERSITÄT BONN

WHAT IS THE PATH INTEGRAL FORMALLY?

The "Path Integral" uses the generalisation of multi-dimensional integrals…

$$\int dx_1 dx_2 dx_3 \cdots dx_n \; f(x_1, x_2, x_3, \cdots, x_n)$$

variables — function of variables

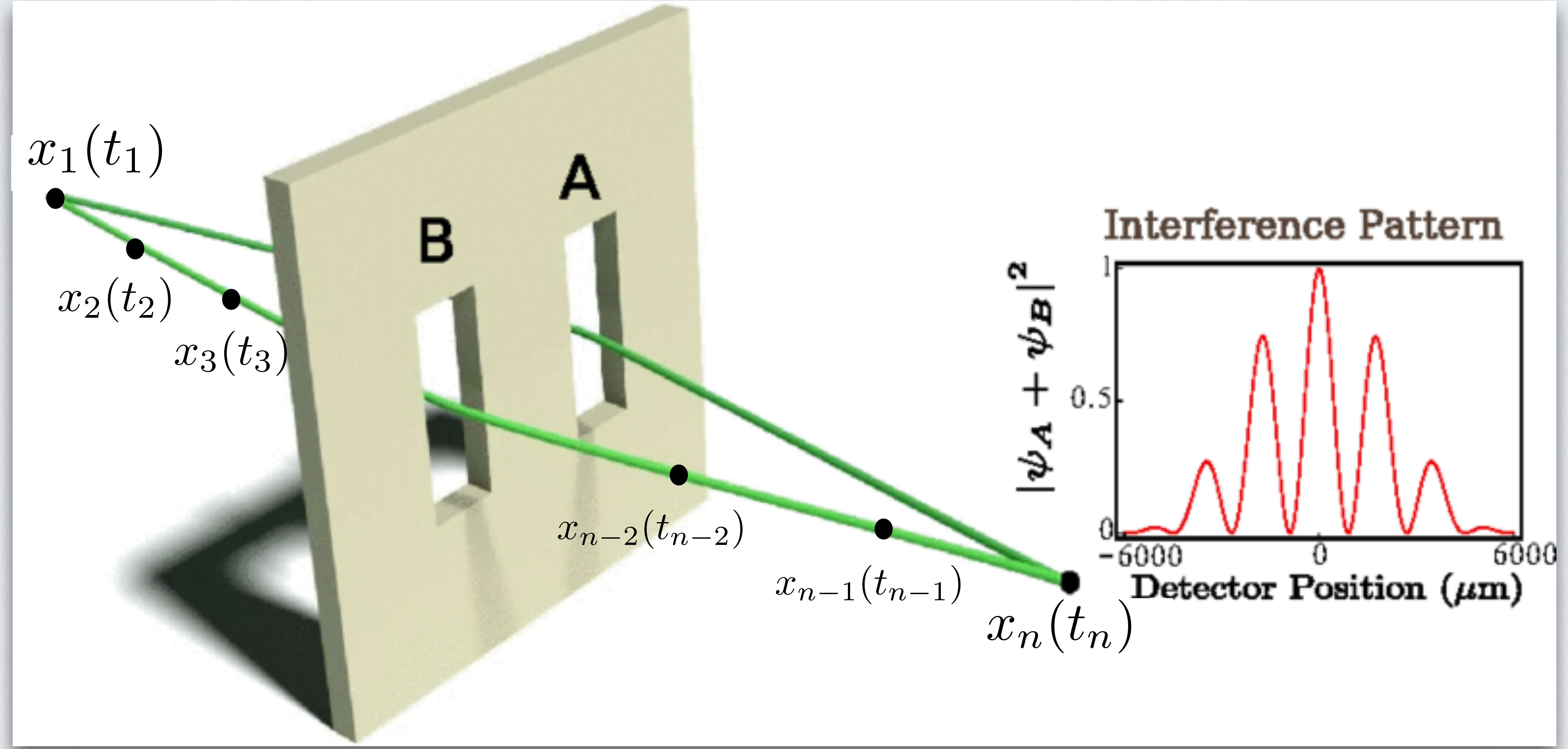
…to a multi-dimensional integral over functions

$$\int dx_1(t_1) dx_2(t_2) \cdots dx_n(t_n) \; f[x_1(t_1), \cdots, x_n(t_n)]$$

functions — function of functions

WHY IS THE PATH INTEGRAL IMPORTANT IN PHYSICS?

Paths come from connecting the points



- Generalizes "action principle" of classical mechanics to quantum mechanics
- Indispensable tool for quantum theories involving fields
- Amenable to computer simulation

BUT FIRST AN HISTORICAL PERSPECTIVE …
DEFINING THE FUNCTIONAL

Vito Volterra

Norbert Wiener

"function" $f(x)$

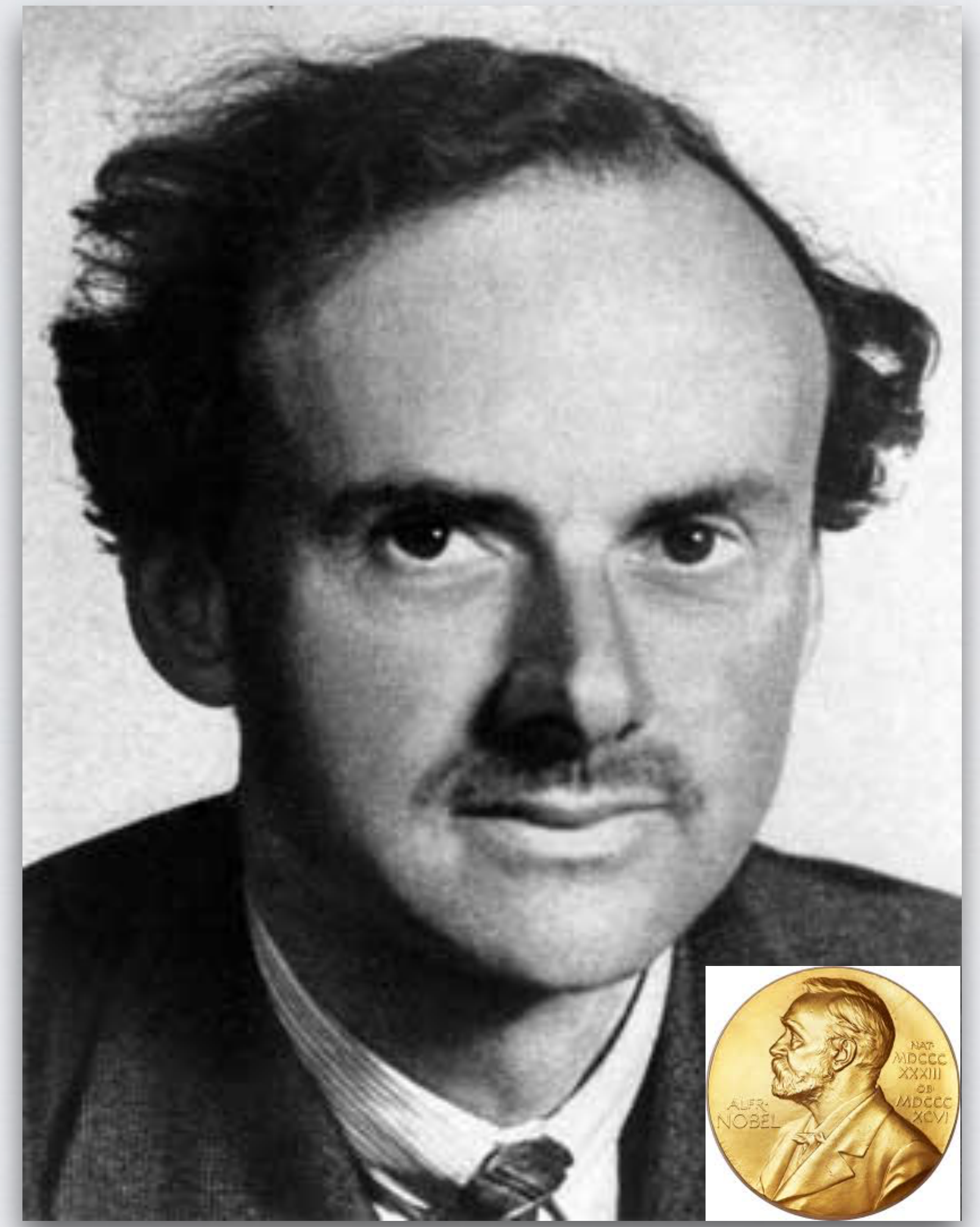
"functional" $f[x(t)]$

"integral" $\int dx\ f(x)$

"Wiener integral" $\int \mathcal{D}[x(t)]\ f[x(t)]$

APPLYING FUNCTIONALS TO PHYSICS

P. A. M. Dirac

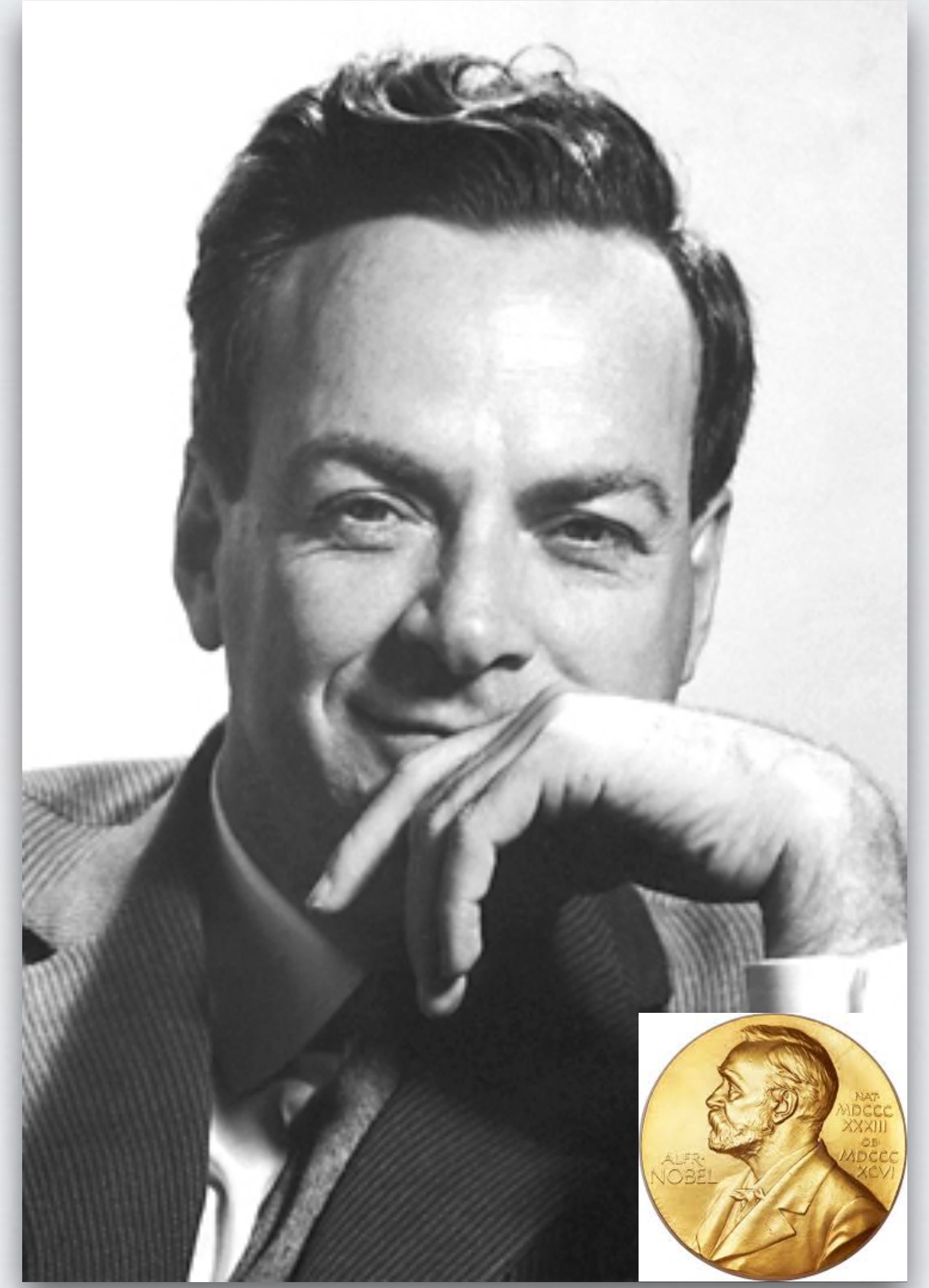


Dirac Equation

$$(i\gamma^{\mu}\partial_{\mu} - m)\,\psi(x,t) = 0$$

1933

Richard Feynman



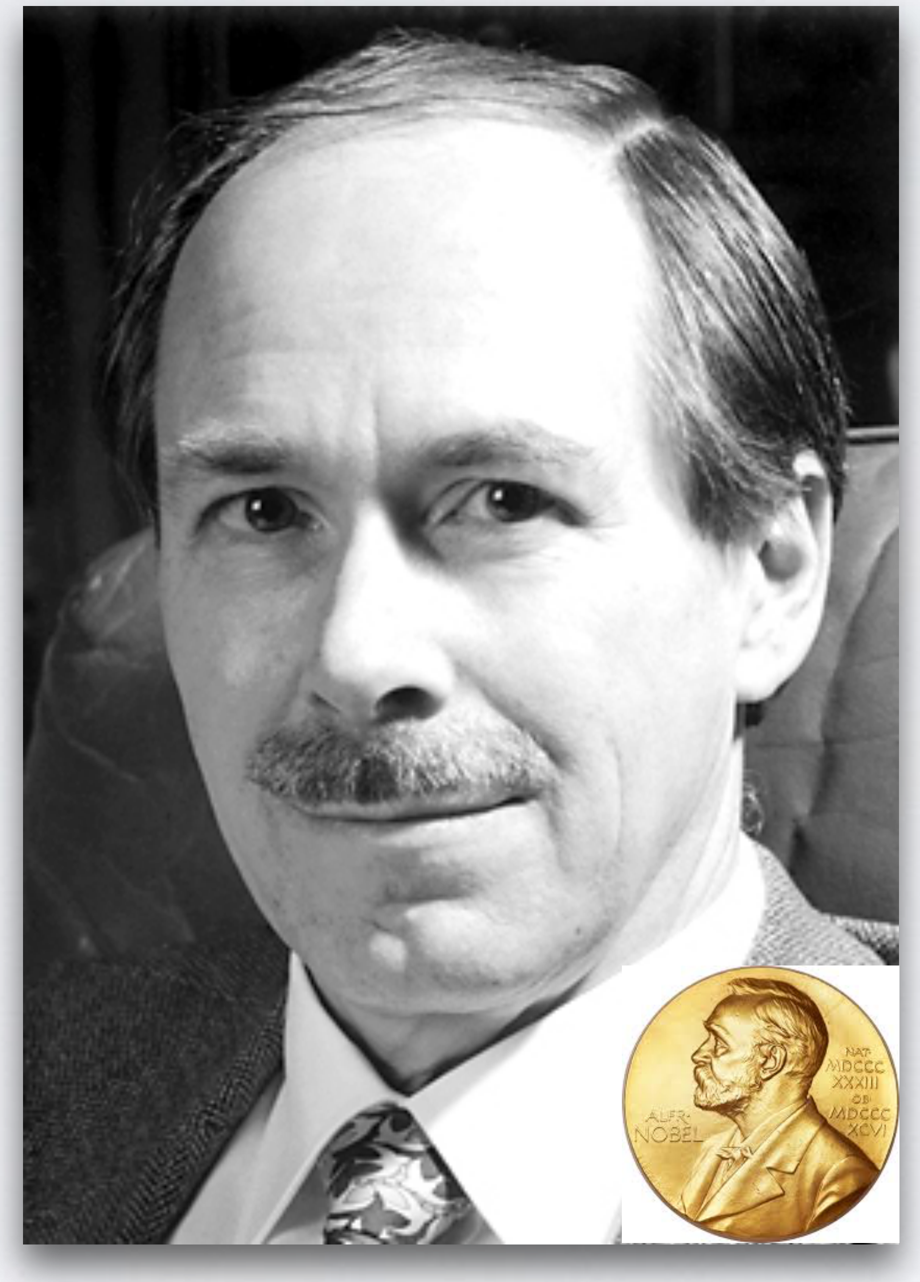
"Probability Amplitude" as a sum over all paths

$$\langle x|e^{-iHT/\hbar}|y\rangle$$

1948

THE PATH INTEGRAL FORMALISM IN THE "MODERN ERA"

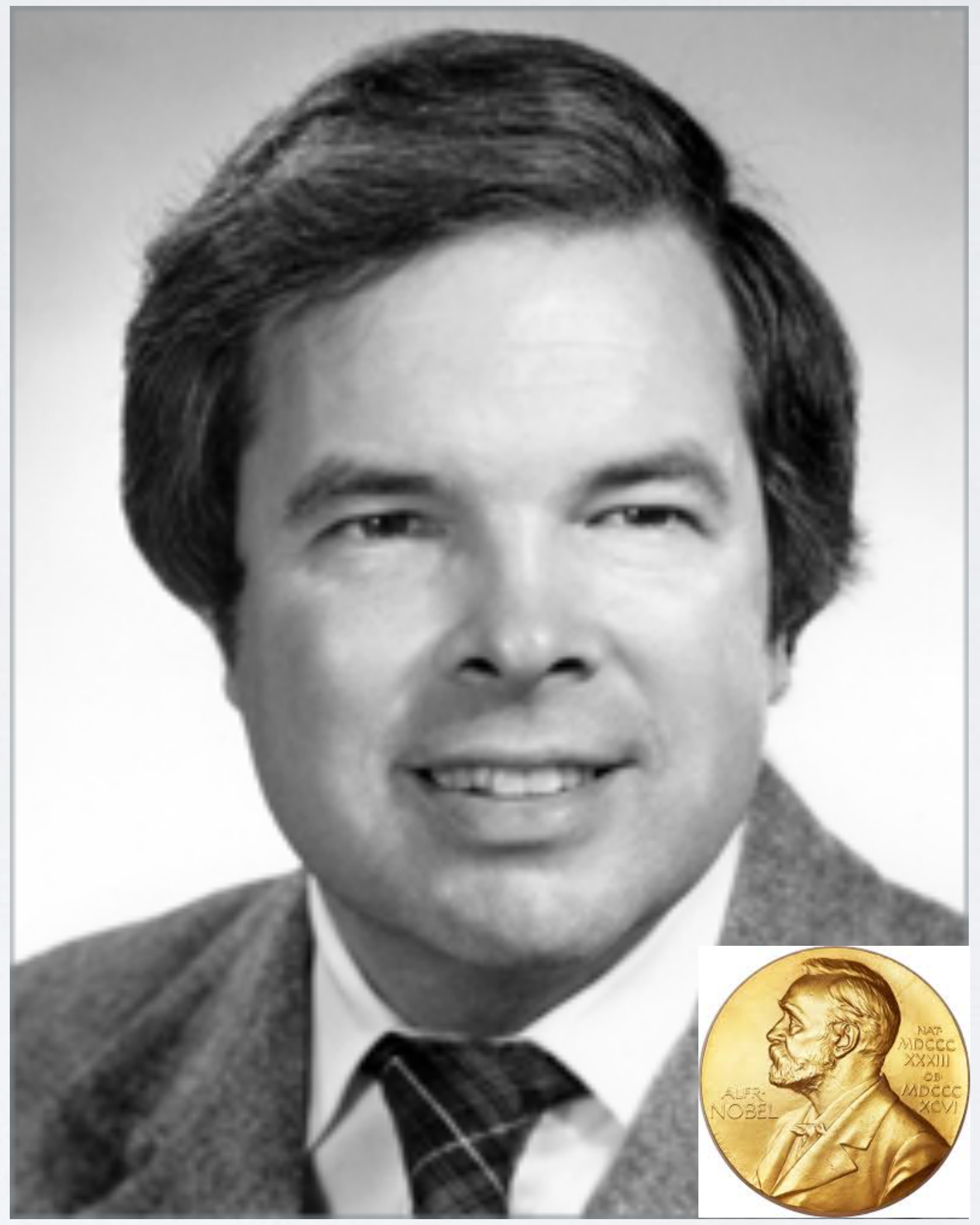
Gerardus 't Hooft



Renormalization of gauge theories using path integral formalism

1971

Kenneth Wilson

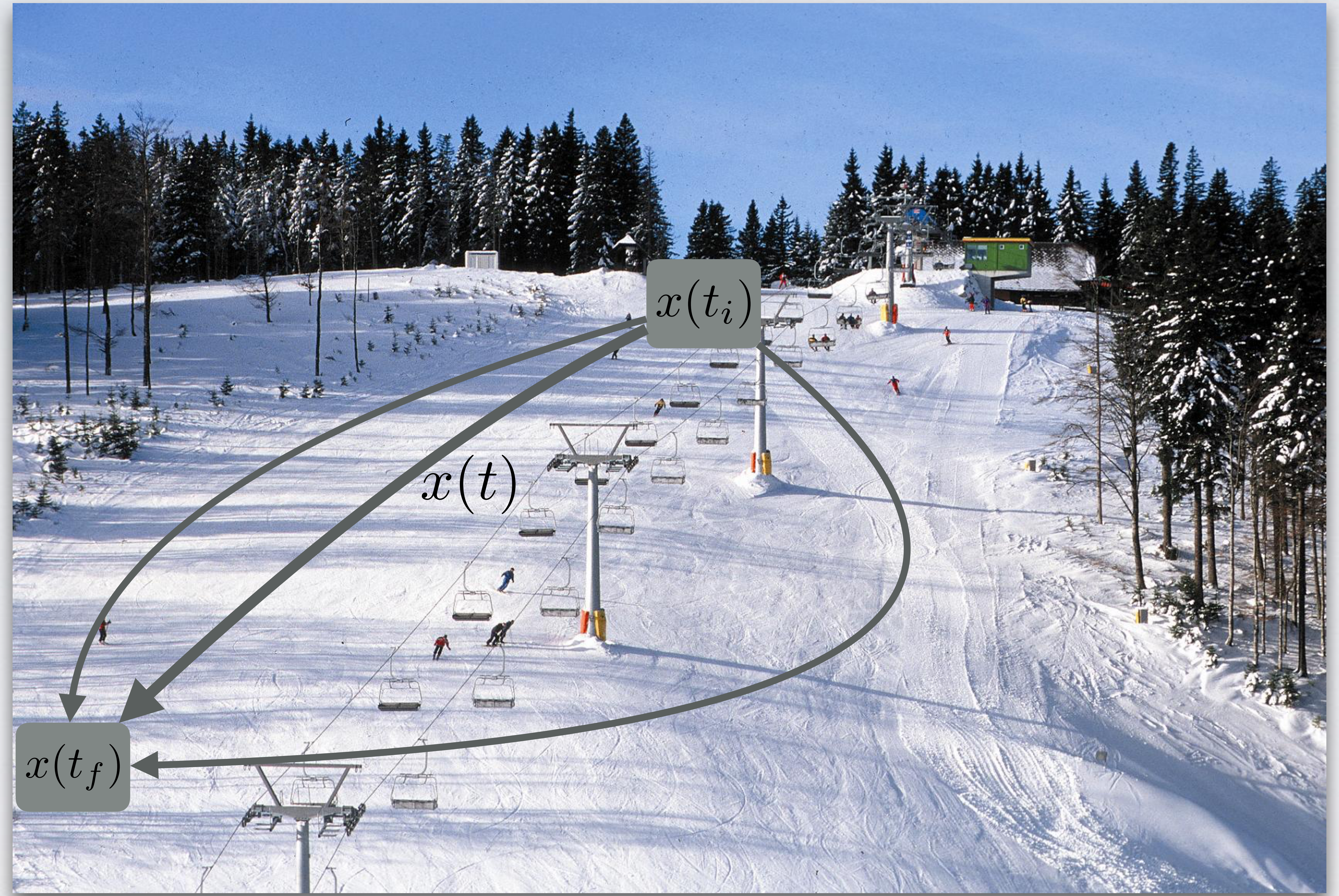


Pioneered the use of computers to calculate physical observables

1970s-1980s

PRINCIPLE OF "LEAST ACTION"

(CLASSICALLY SPEAKING…)



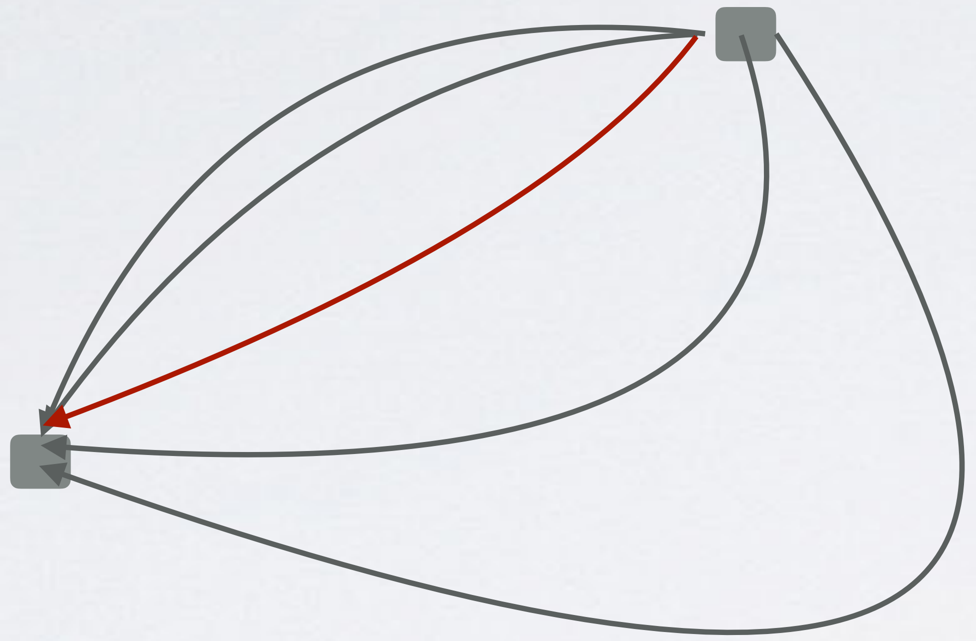
"action"

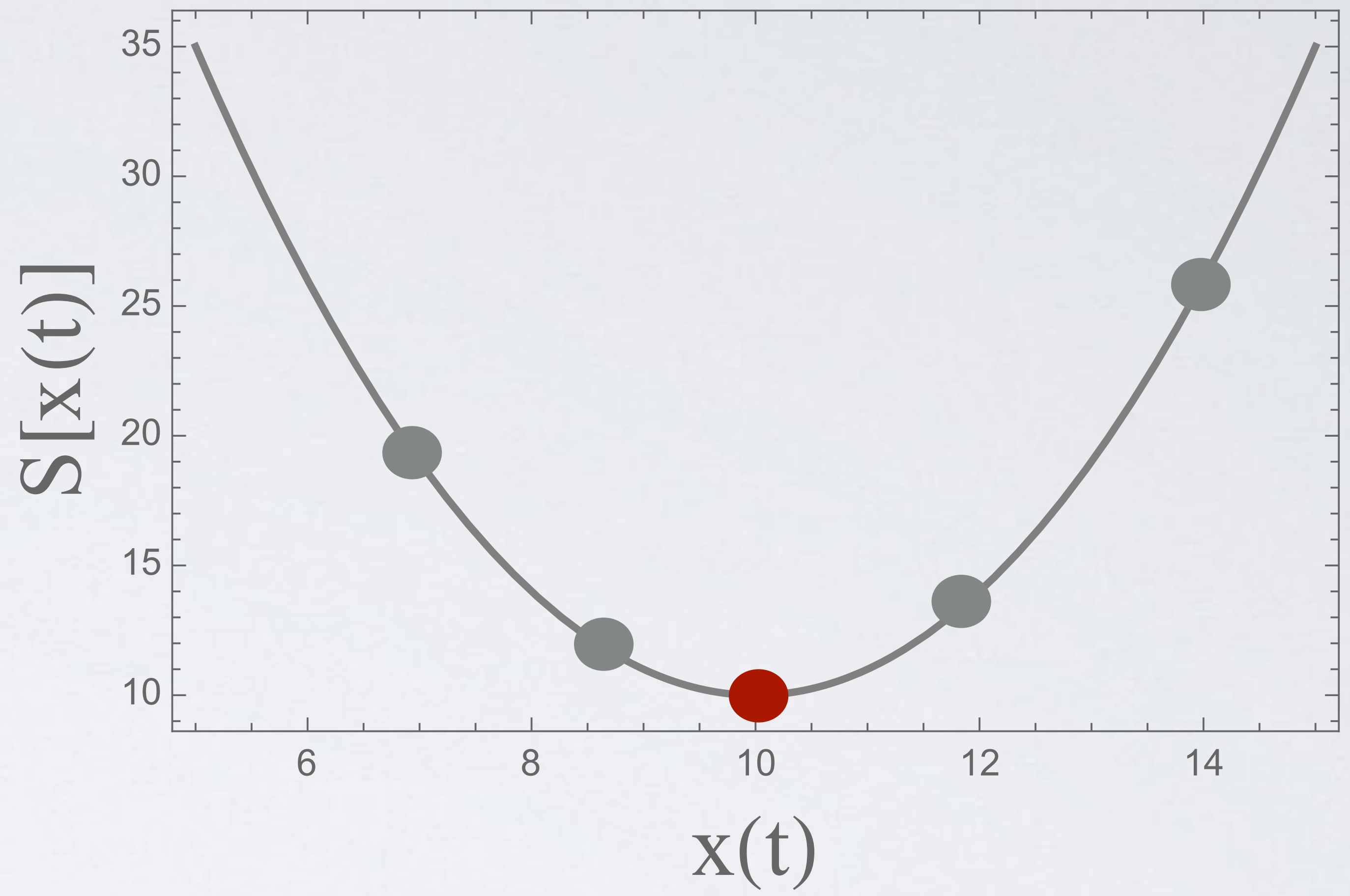
$$S[x(t)] = \int_{t_i}^{t_f} dt \; \mathcal{L}[x(t), \dot{x}(t), t]$$

The "true" path is the path that minimises S

HOW DO YOU DETERMINE THE "TRUE" PATH?

Brute Force Method:
Sample ***millions*** of paths





Eloquent Method:
Use our brains

$$S[x(t)] = \int_{t_i}^{t_f} dt\ \mathcal{L}[x(t), \dot{x}(t), t] = \int_{t_i}^{t_f} dt\ \left[\frac{m}{2}\dot{x}(t)^2 - V(x(t))\right]$$

k.e. p.e.

stationary ~~minimum~~ $\Longrightarrow \delta S[x(t)] = 0$

$$\Longrightarrow m\ \ddot{x}(t) = -V'(x(t))$$

$$ma = F$$

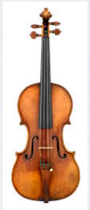
Newton's 2nd Law of Motion

Principle of "Stationary" Action

AN ALTERNATIVE METRIC: $P[x(t)] = \frac{\exp(-S[x(t)]/\hbar)}{\int \mathcal{D}[y(t)]\,\exp(-S[y(t)]/\hbar)}$

THE PATH WITH THE LARGEST VALUE OF "P[ ]" IS THE "TRUE" PATH

(CLASSICALLY SPEAKING)

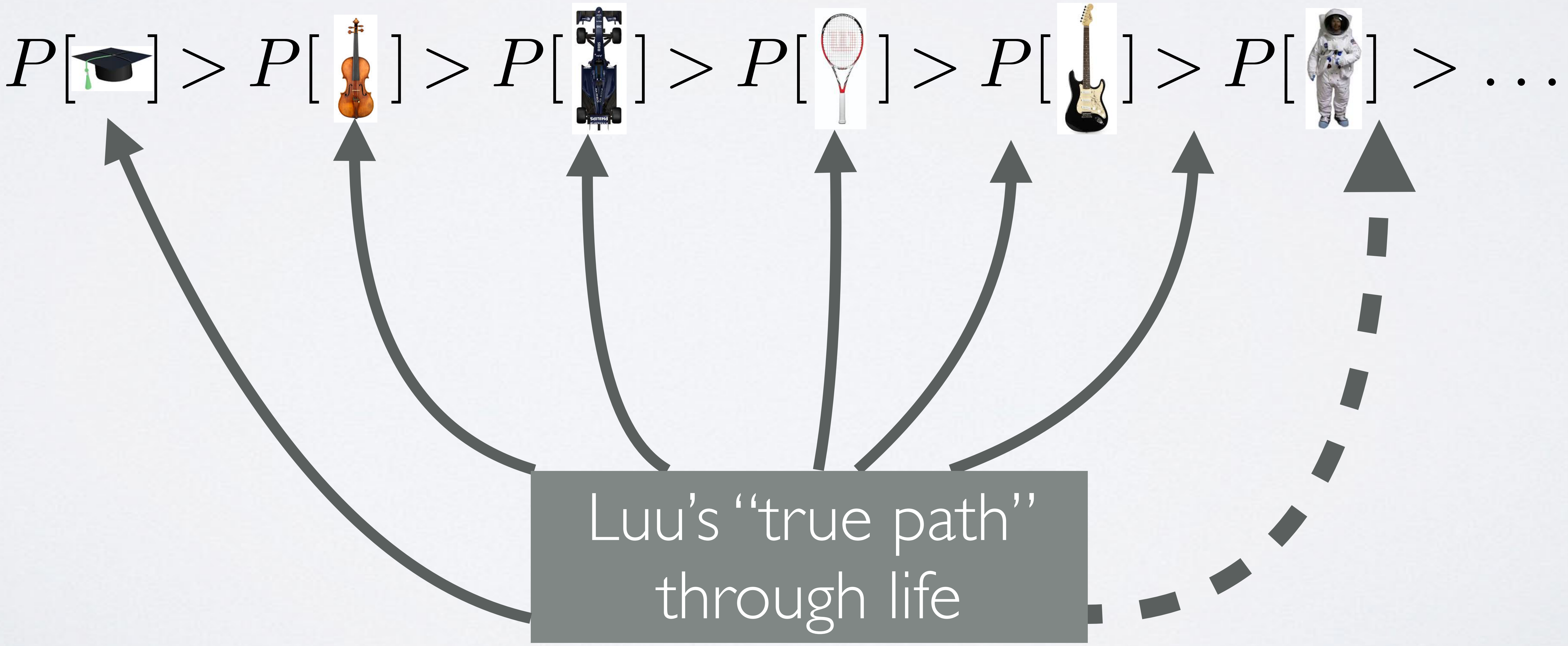
$P[$$] > P[$$] > P[$$] > P[$$] > P[$$] > P[$$] > \dots$

Luu's "true path"
through life

THE PATH INTEGRAL AT THE QUANTUM SCALE

INTERPRET "P[]" AS A PROBABILITY

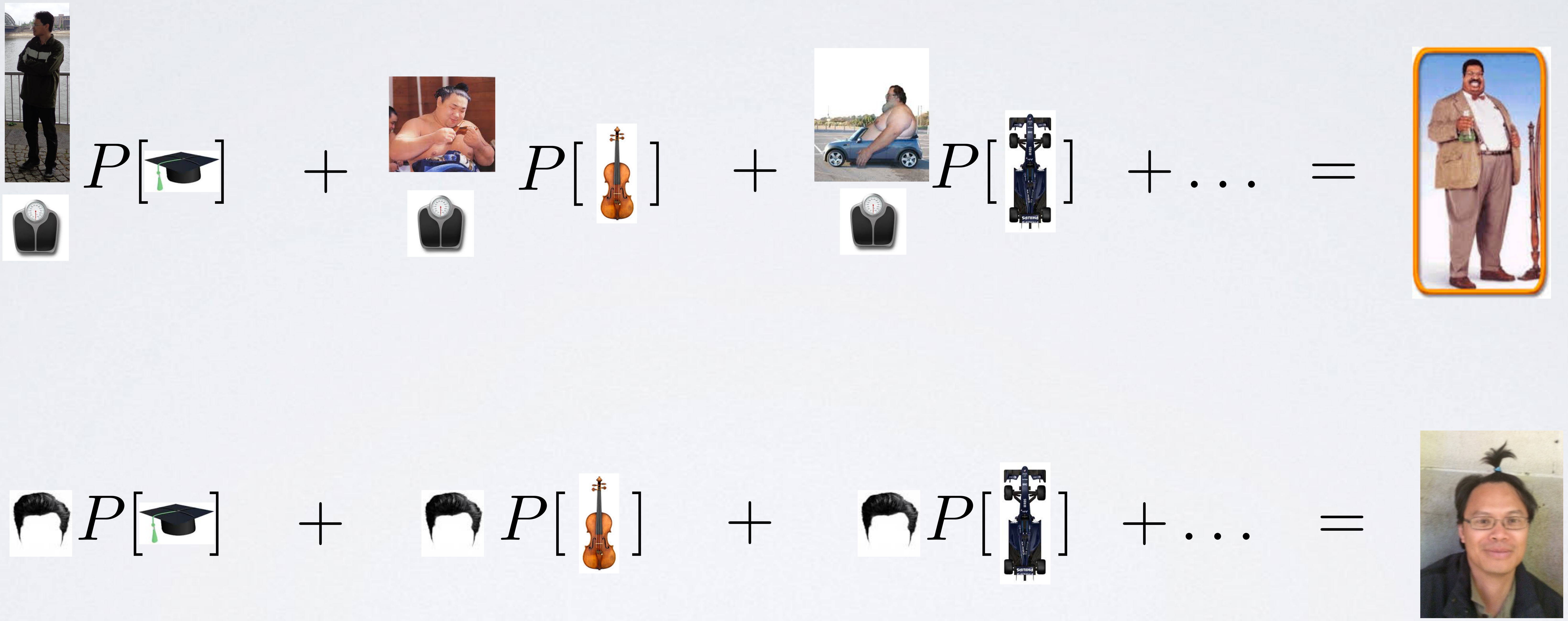
$$P[x(t)] = \frac{\exp(-S[x(t)]/\hbar)}{\int \mathcal{D}[y(t)] \exp(-S[y(t)]/\hbar)}$$



LUU'S "QUANTUM PATH(S)" THROUGH LIFE

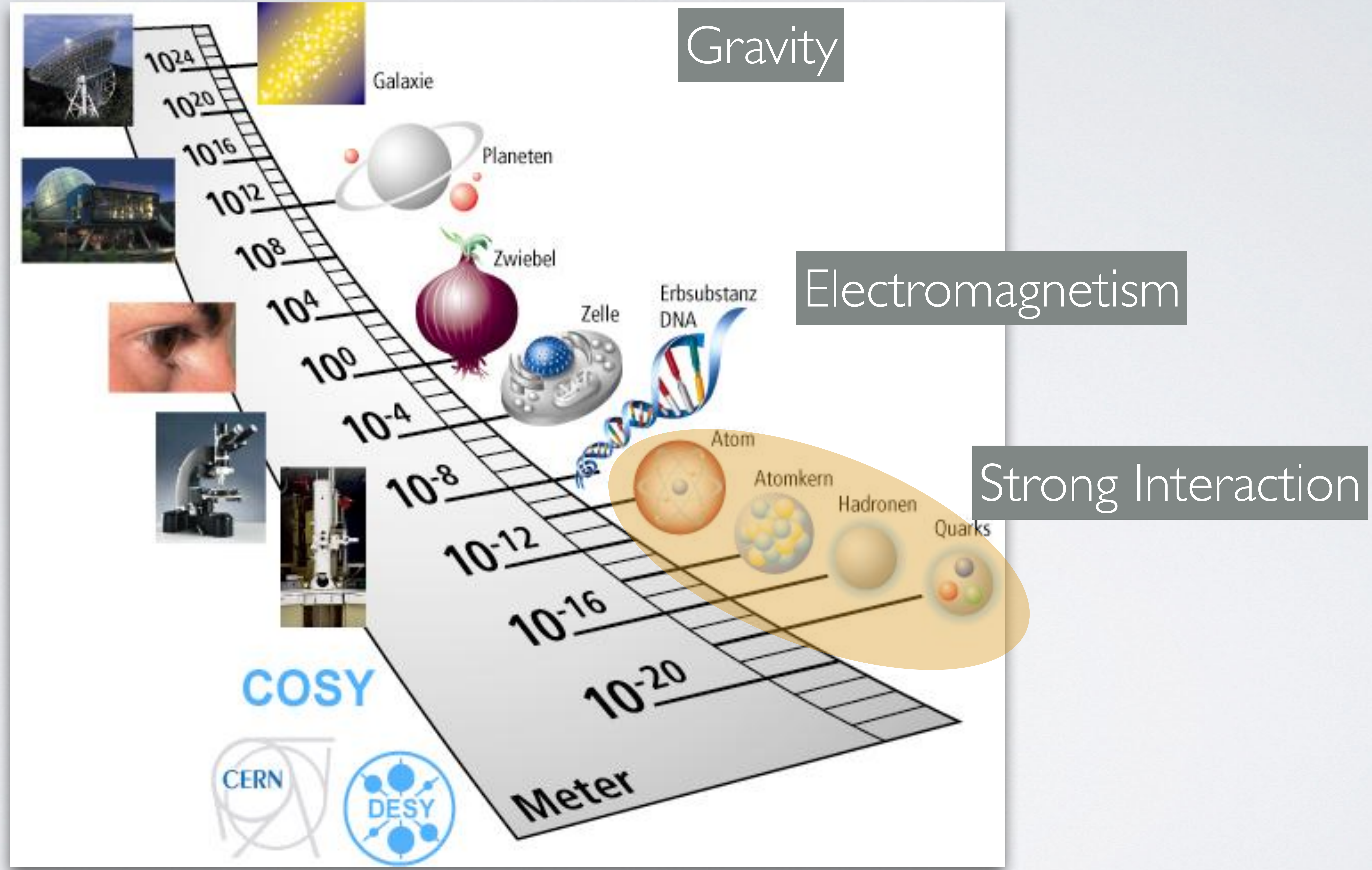
PERFORM "WEIGHTED" AVERAGE OF MY WEIGHT

(QUANTUM MECHANICALLY SPEAKING)



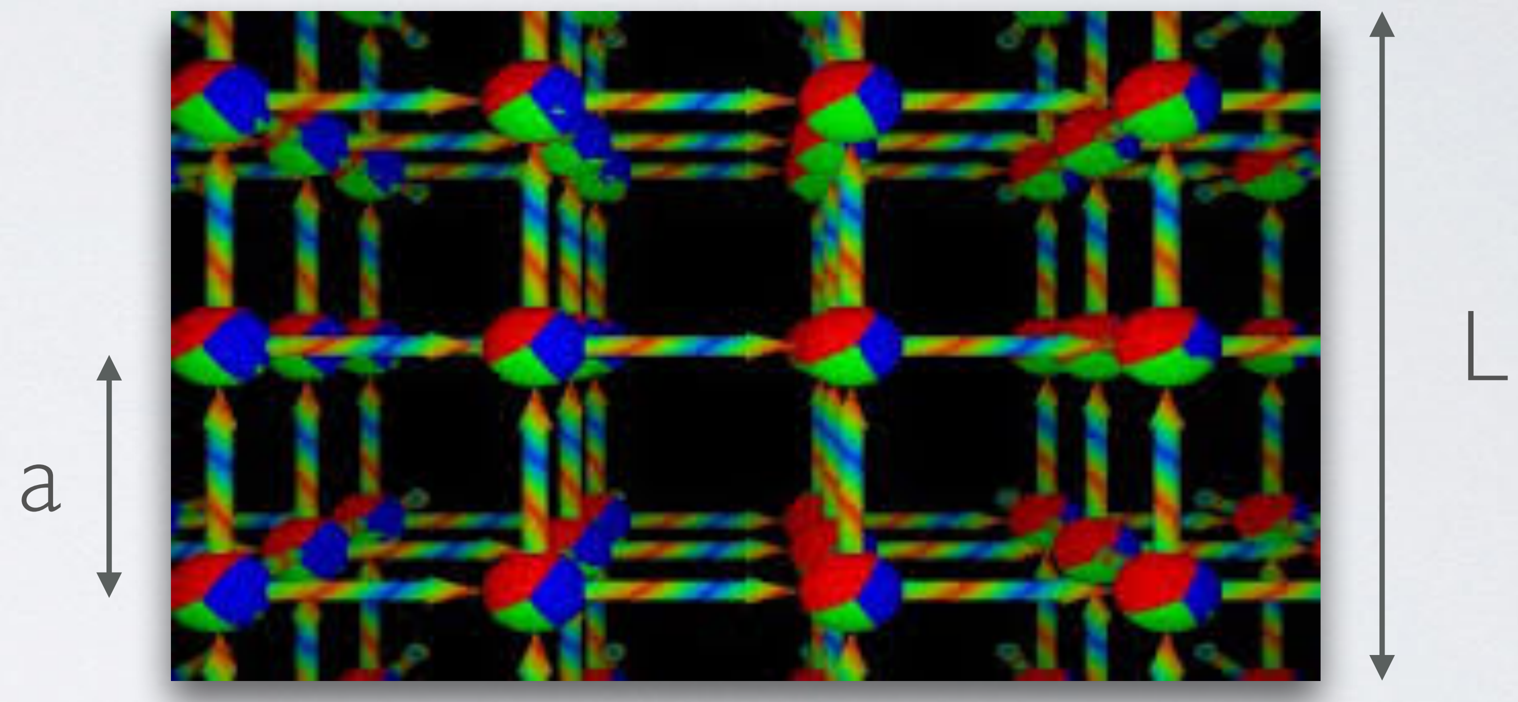
We can obtain information about the system with appropriate probes

A LITTLE PERSPECTIVE IN SCALES…

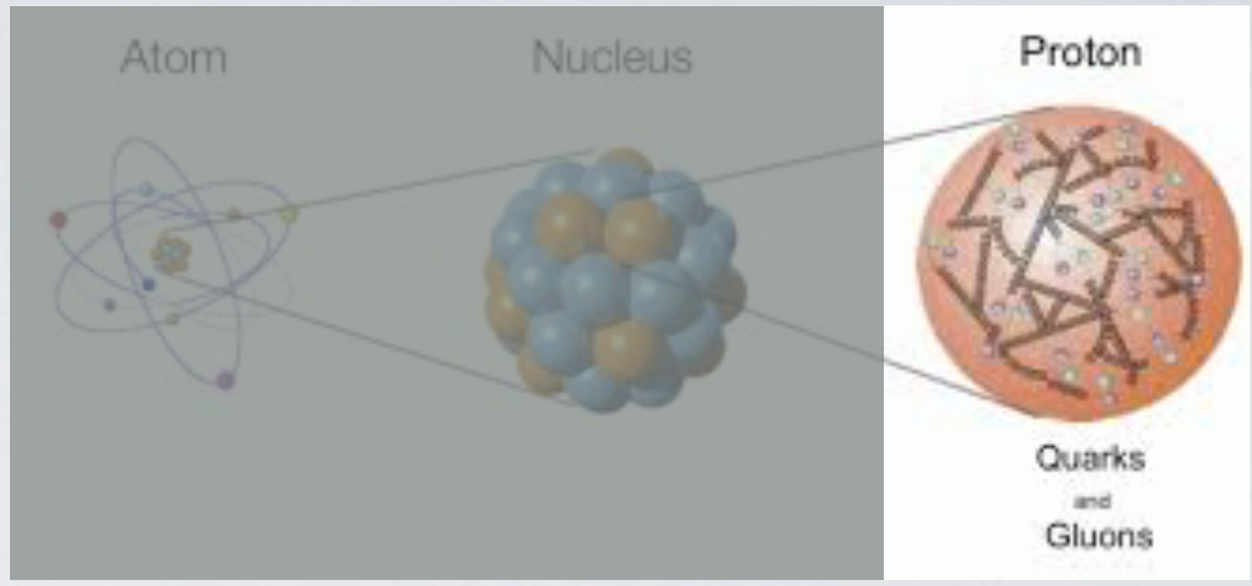


THE ROLE OF HIGH-PERFORMANCE COMPUTING

- Discretise space and time onto a lattice
- Reformulate theory on discretised space
- “Randomly” sample paths on the discretised lattice
  - Not all paths created equally — need to be clever on how to sample paths
  - Stochastic — measurements have statistical uncertainty
- Take $a \to 0$ continuum limit
  $L \to \infty$ infinite volume limit



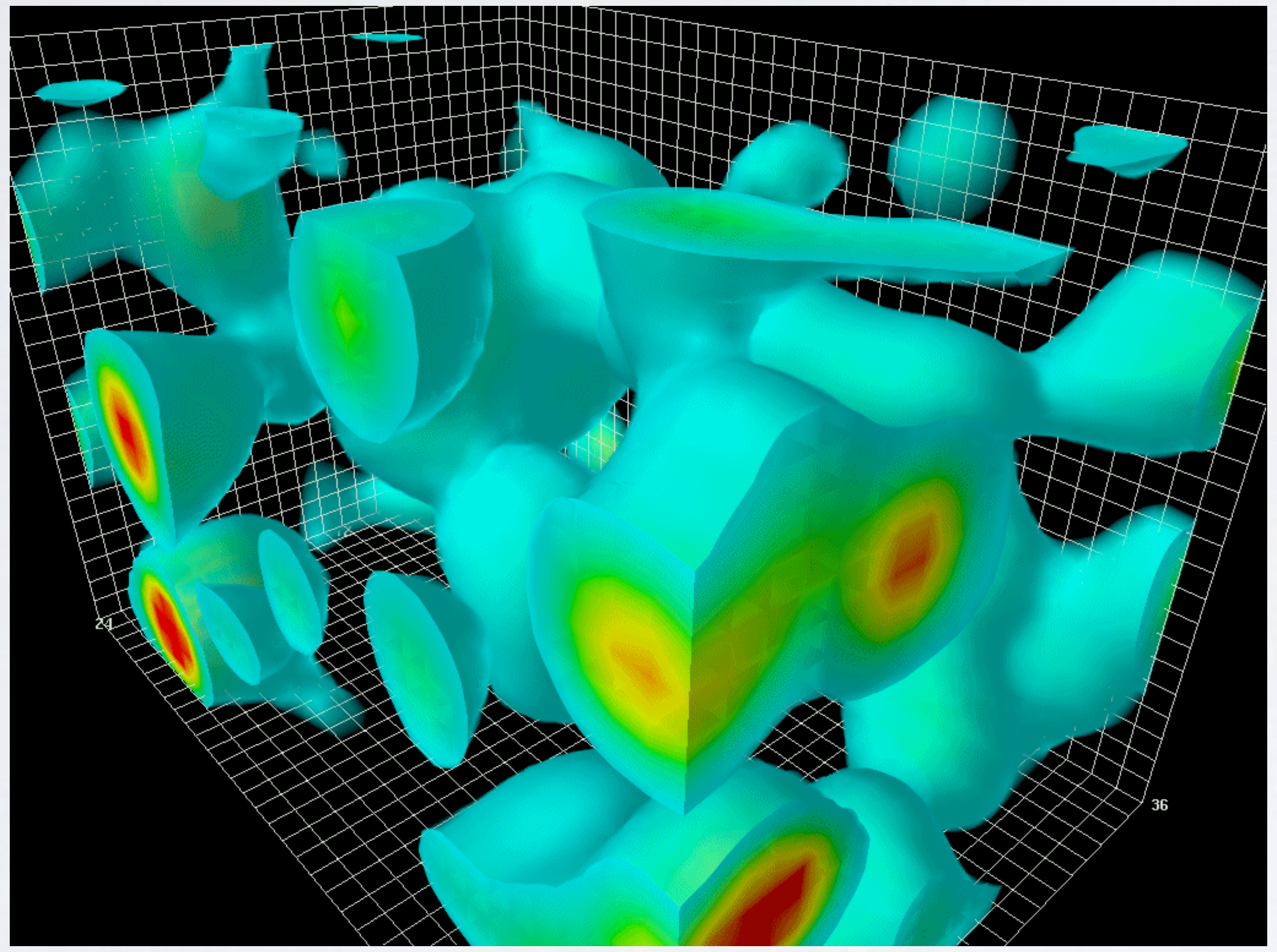
QUARKS AND GLUONS



Lattice Quantum Chromodynamics (LQCD): simulating quarks and gluons on a space-time lattice

$L \sim 4 - 6\ \mathrm{fm}$

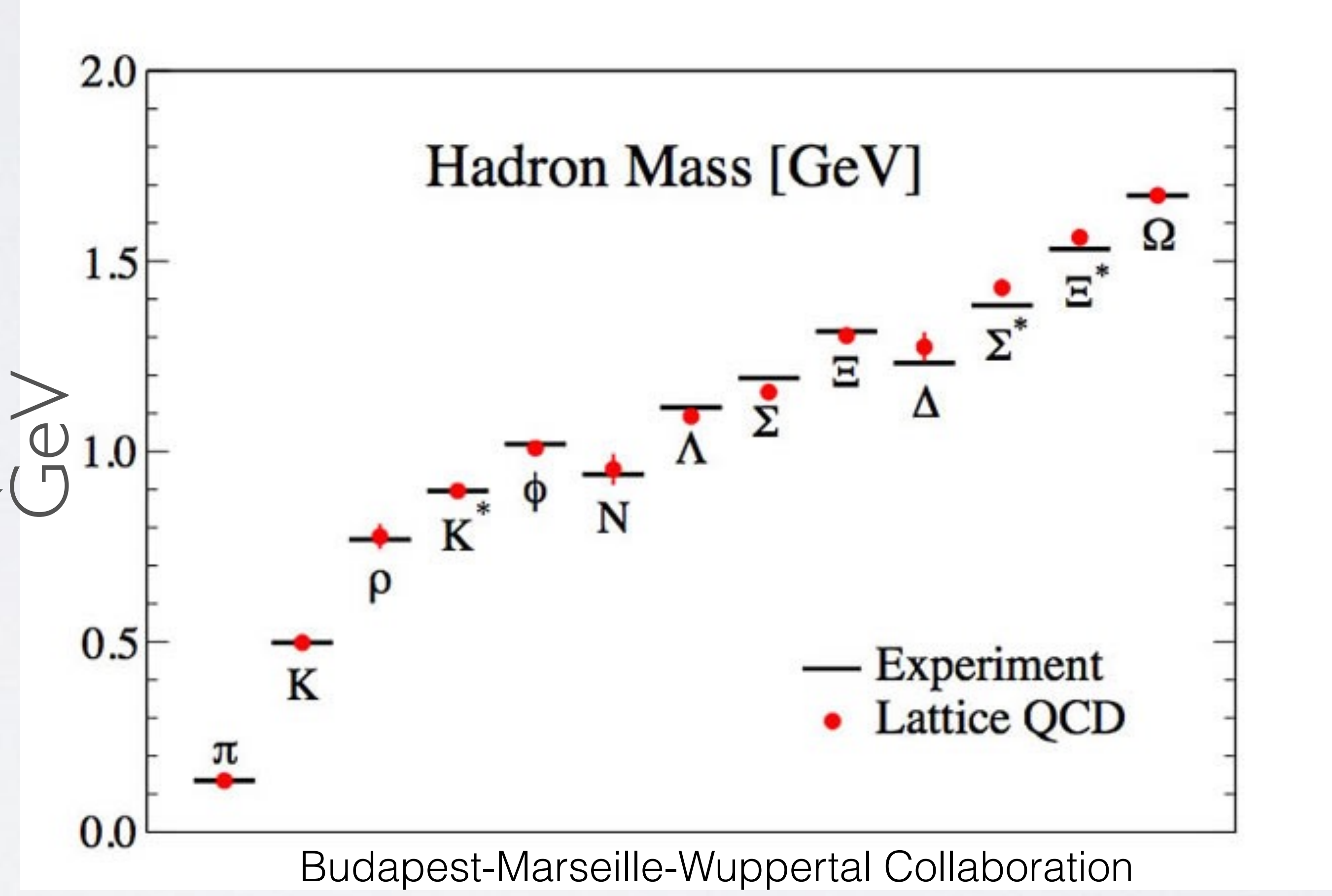
$a \sim .1\ \mathrm{fm}$



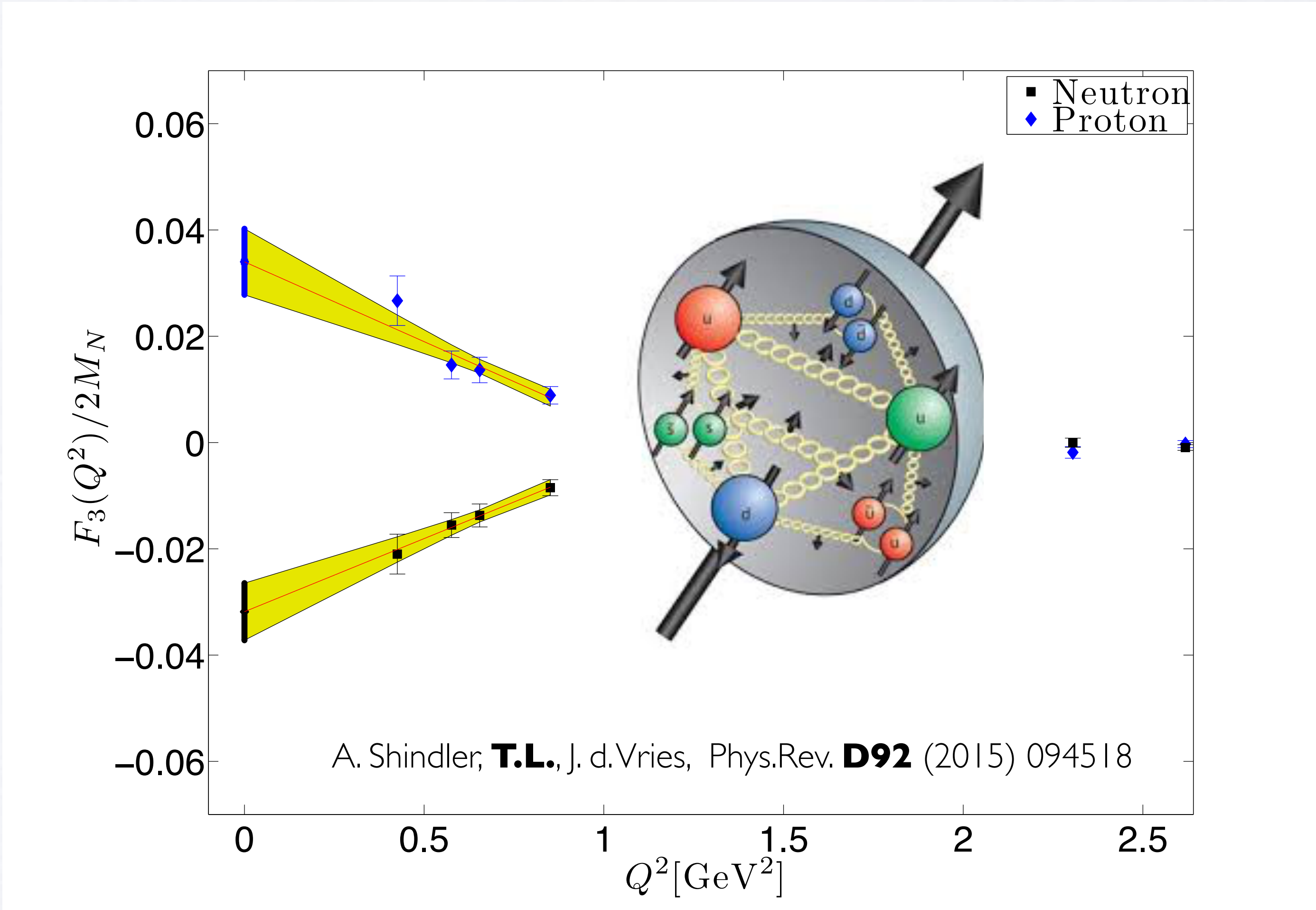
Quantum fluctuations of gluonic fields

Movie courtesy of D. Leinweber
http://www.physics.adelaide.edu.au/theory/staff/leinweber/VisualQCD/Nobel/

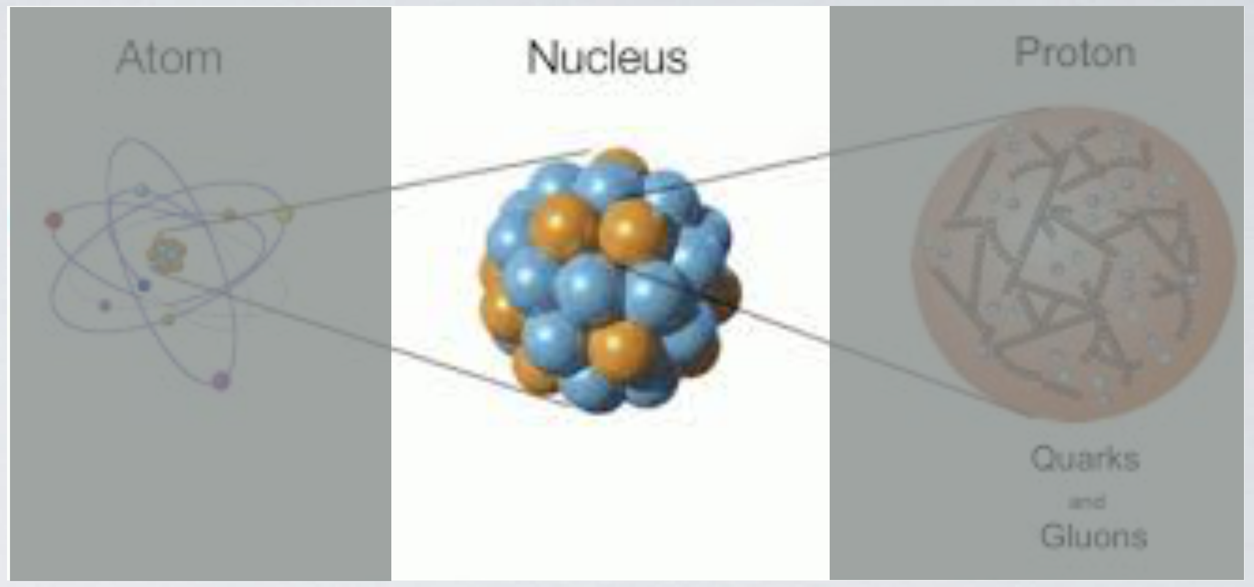
Origin of hadron masses



Nucleon electric dipole moment



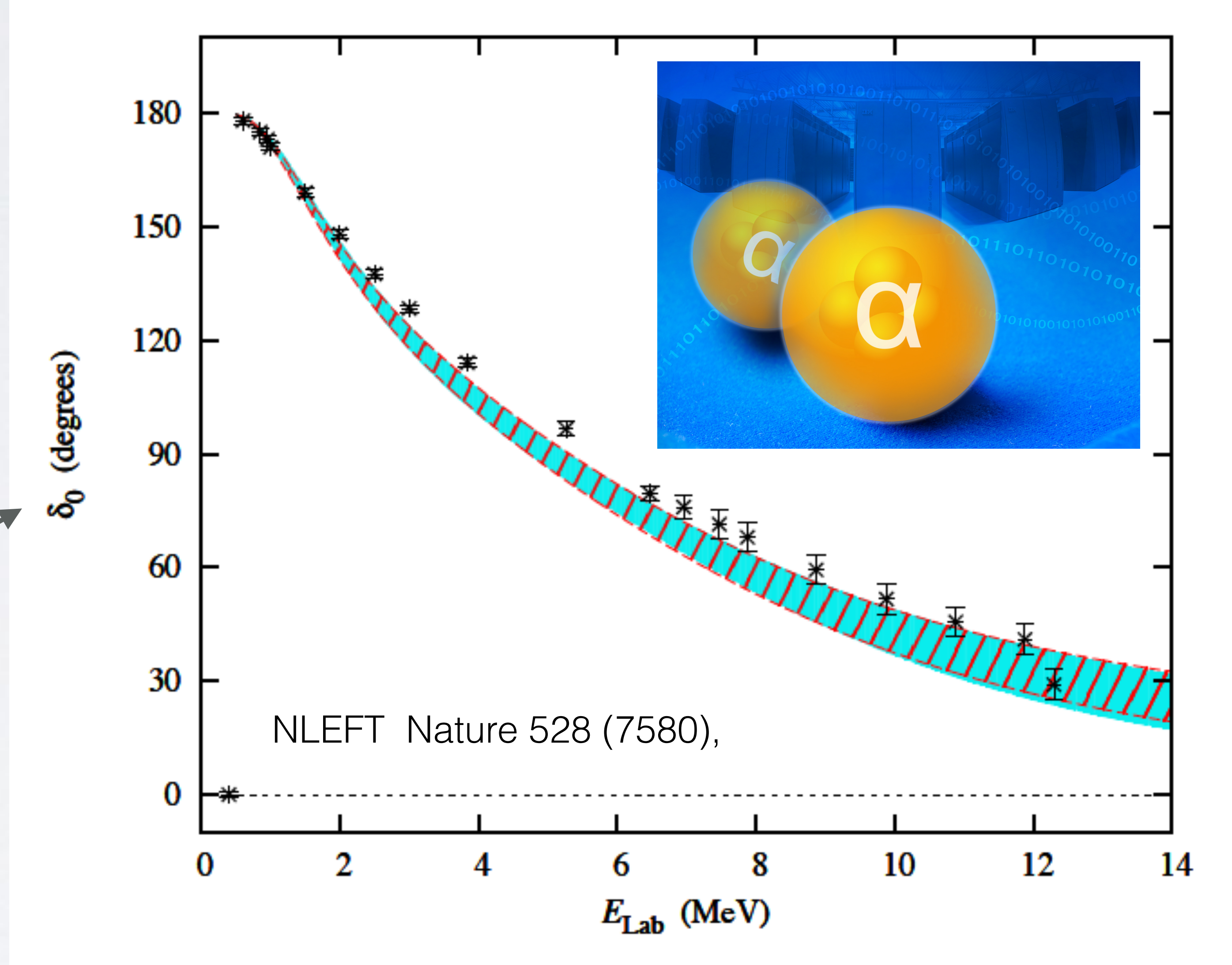
HADRONIC SYSTEMS



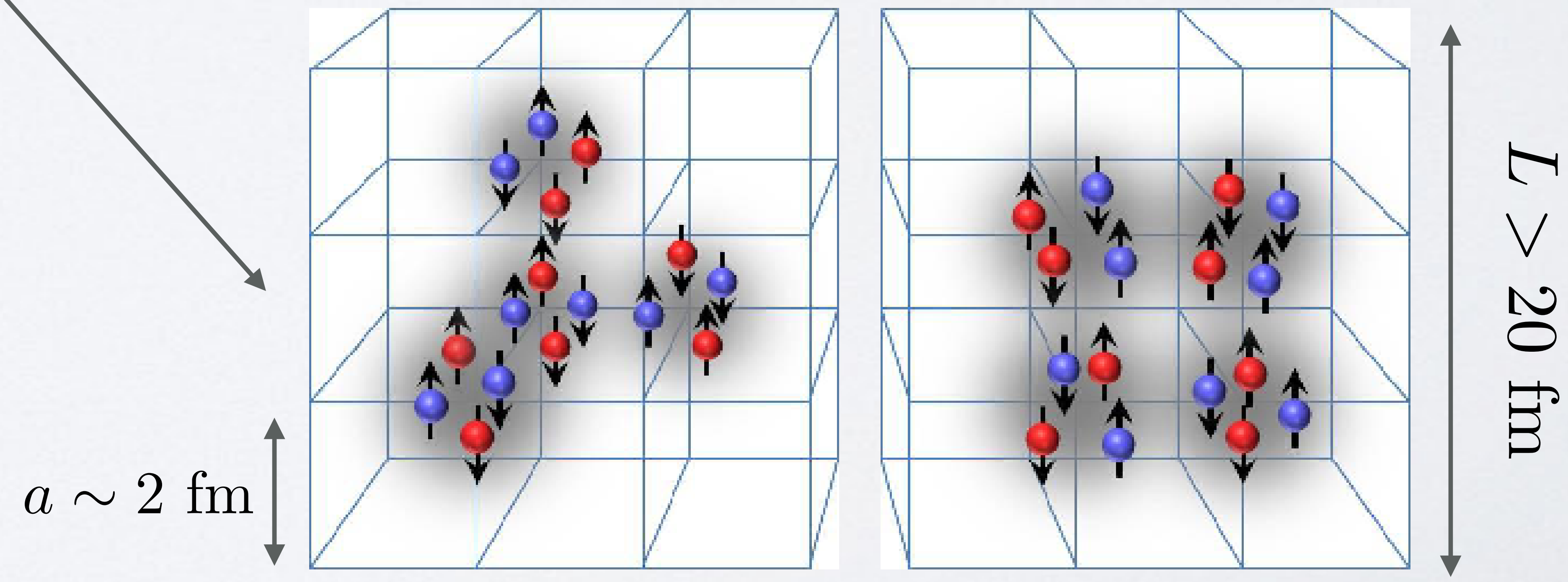
Nuclear Lattice Effective Field Theory (NLEFT):
nucleons (protons and neutrons) are degrees of freedom on a discretised space/time lattice

Triple-alpha process in heavy stars

Movie courtesy of FZJ Supercomputing Centre



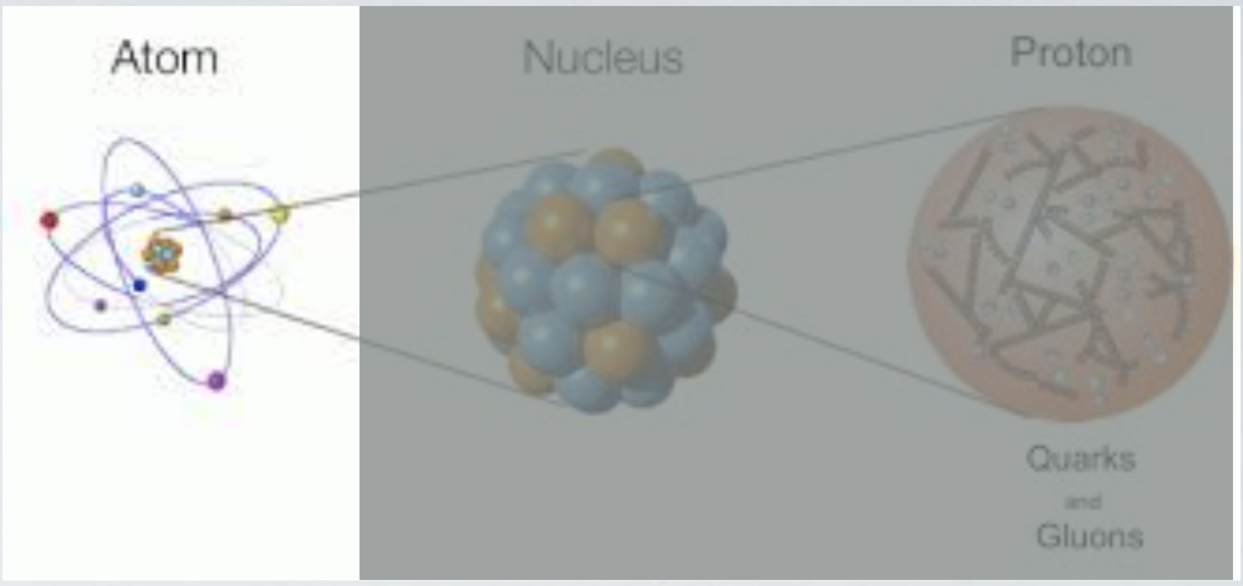
Alpha clustering in Carbon



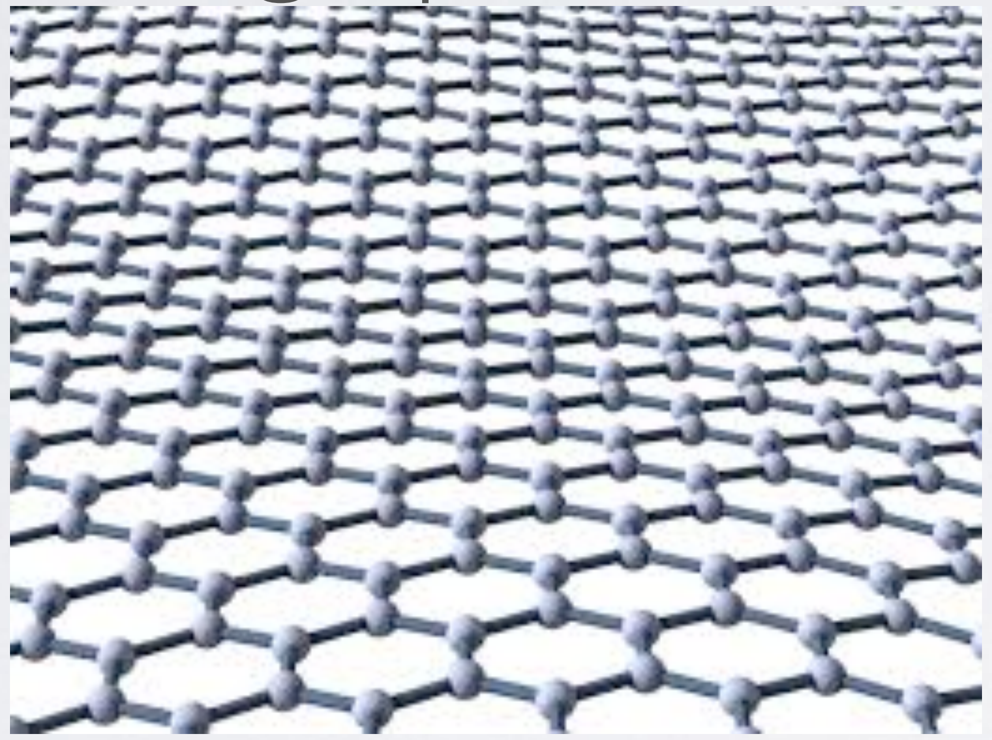
NLEFT Phys.Rev.Lett. **112**

STRONGLY CORRELATED ELECTRONS

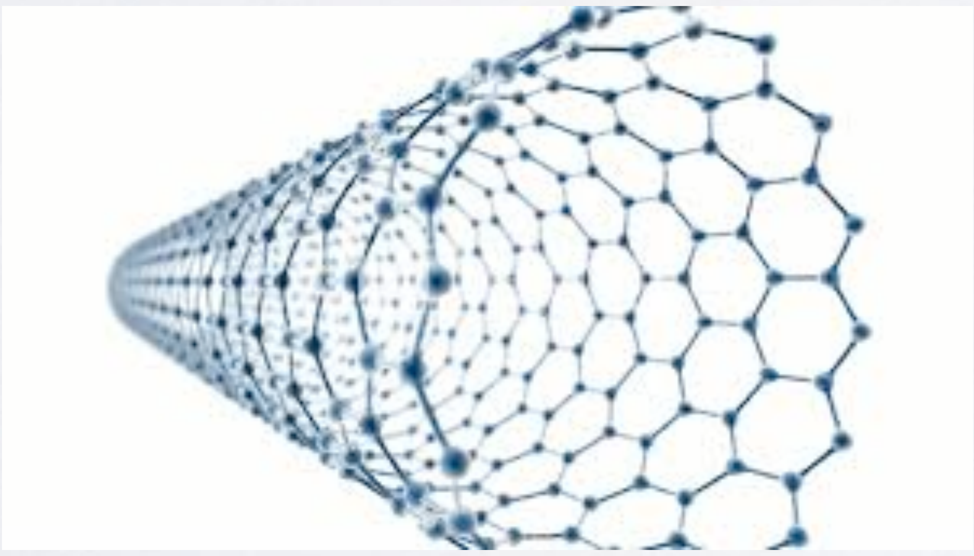
Simulating electrons on a hexagonal lattice



graphene



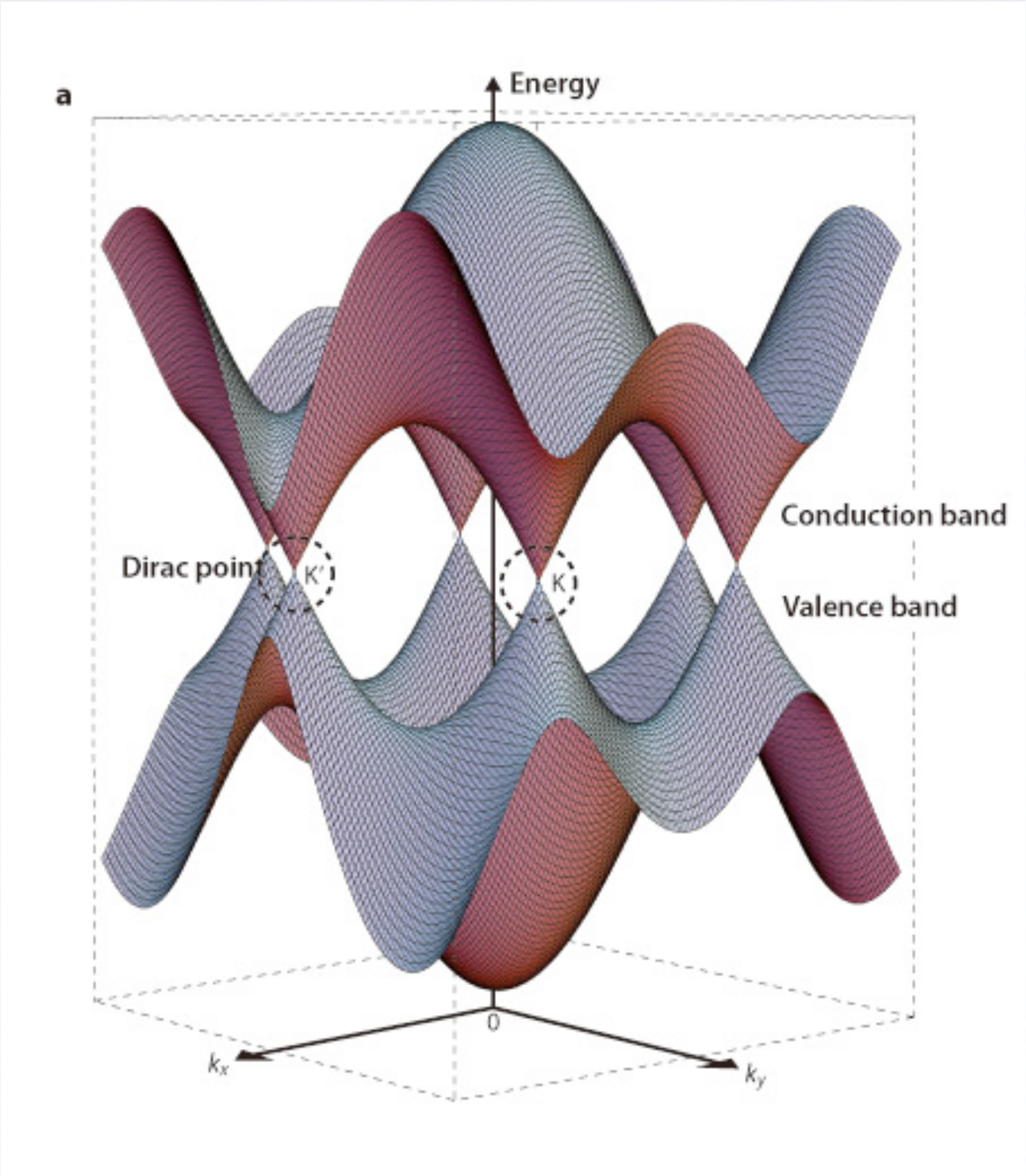
nanotube



$L > 1$ nm $\quad a = 1.42$ Å

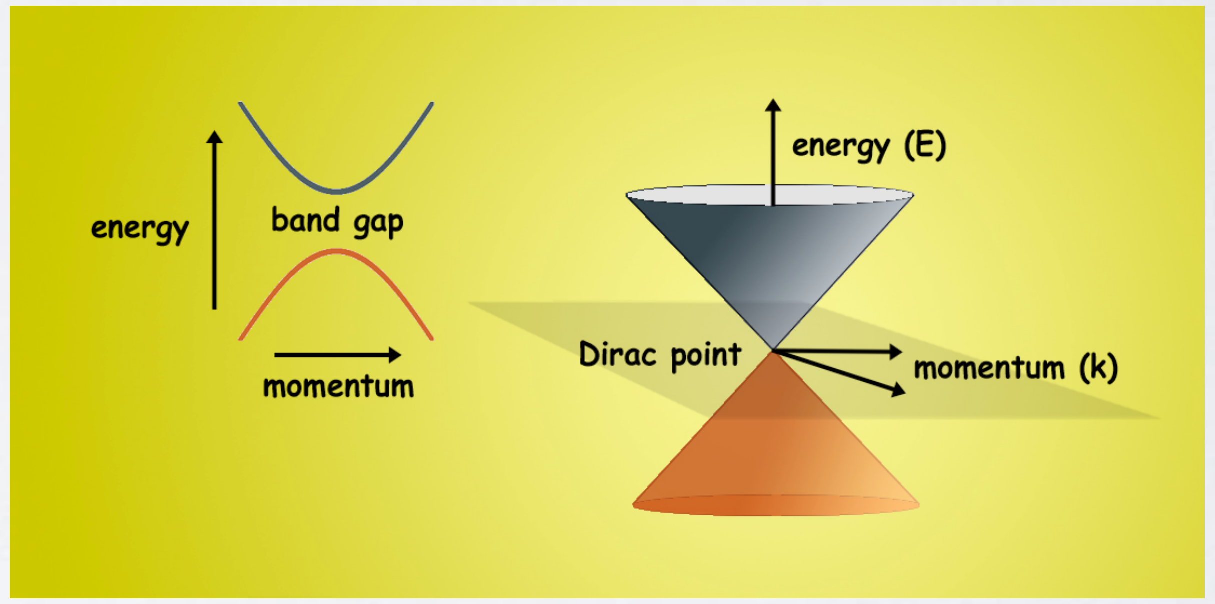
Carbon nanostructures

Dispersion relation



Tight-binding approximation

Interaction-induced Mott gap?



T.L., T. Lahde, arXiv:1511.04918

w/ Coulomb interaction

LET'S MAKE THINGS A LITTLE MORE FORMAL: EUCLIDEAN PATH INTEGRAL

- First we ***Wick*** rotate to Euclidean time

$$\tau \to i\tau$$

- Given a ***time-dependent, local*** Hamiltonian,

$$H(\tau) = \frac{p^2}{2m} + V(x,\tau)$$

$$[x_\alpha, p_\beta] = i\hbar\delta_{\alpha,\beta}$$

- The solution to the evolution operator, ***U(t',t)***, is given by Schrödinger's equation

$$\hbar\frac{\partial U(\tau',\tau)}{\partial \tau'} = -H(\tau')U(\tau',\tau)$$

$$\tau' \geq \tau$$

- Formally, matrix elements of ***U(t',t)*** are equivalent to

$$\langle \boldsymbol{x}_f, \tau' | \boldsymbol{x}_i, \tau\rangle = \langle \boldsymbol{x}_f | U(\tau',\tau) | \boldsymbol{x}_i\rangle = \int_{x(\tau)=\boldsymbol{x}_i}^{x(\tau')=\boldsymbol{x}_f} [d\boldsymbol{x}(\tau)]\, e^{-S[\boldsymbol{x}(\tau)]}$$

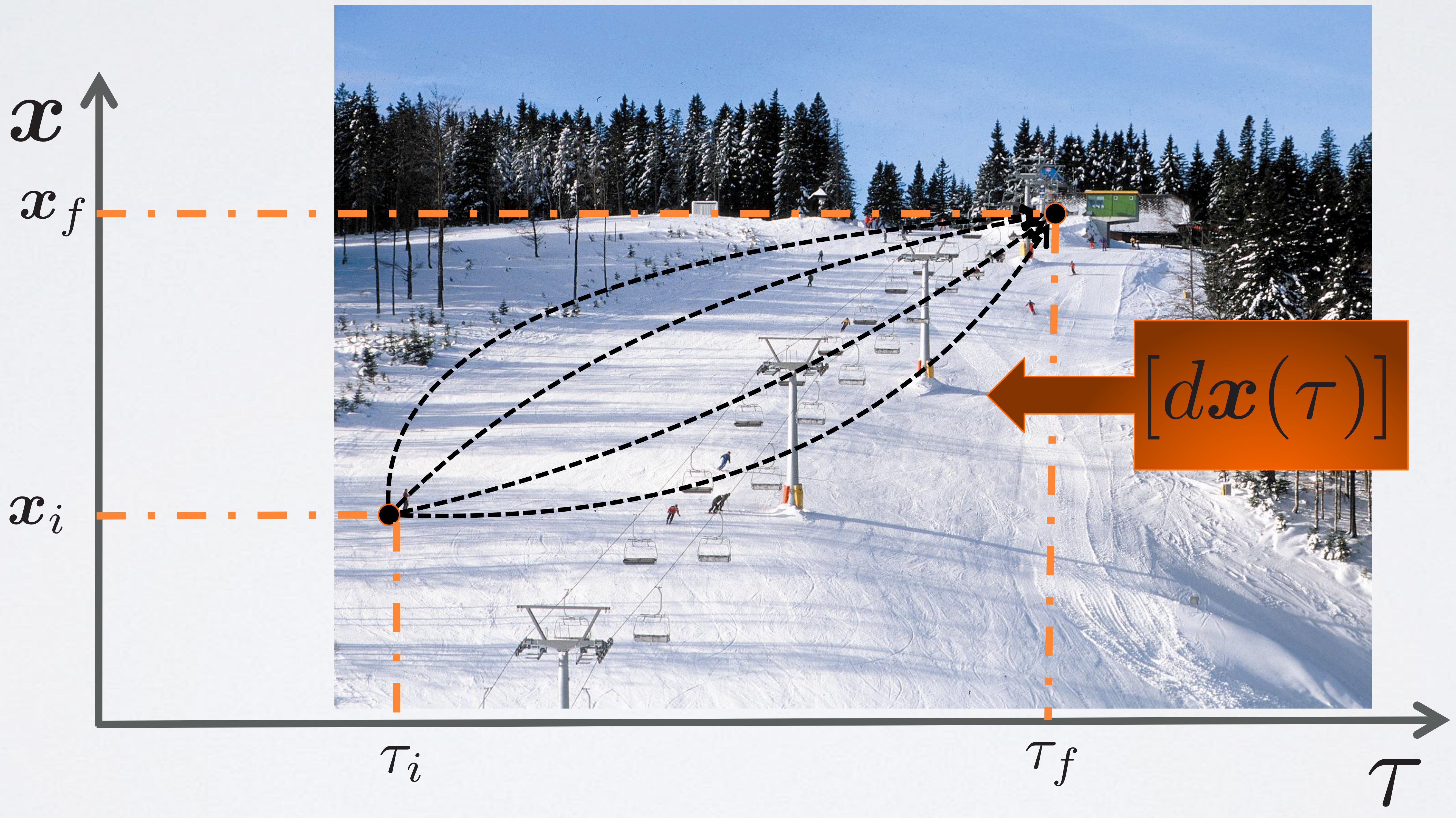
$$S[\boldsymbol{x}(\tau)] = \int_{\tau}^{\tau'} d\tau \left(\frac{\dot{\boldsymbol{x}}(\tau)^2}{2m} + V(\boldsymbol{x}(\tau),\tau)\right)$$

$$\hbar, c \to 1$$

WHAT DOES THIS EXACTLY MEAN?

$$\langle \boldsymbol{x}_f | U(\tau', \tau) | \boldsymbol{x}_i \rangle = \int_{x(\tau)=\boldsymbol{x}_i}^{x(\tau')=\boldsymbol{x}_f} [d\boldsymbol{x}(\tau)]\, e^{-S[\boldsymbol{x}(\tau)]}$$

$$S[\boldsymbol{x}(\tau)] = \int_{\tau}^{\tau'} d\tau \left(\frac{\dot{\boldsymbol{x}}(\tau)^2}{2m} + V(\boldsymbol{x}(\tau), \tau) \right)$$

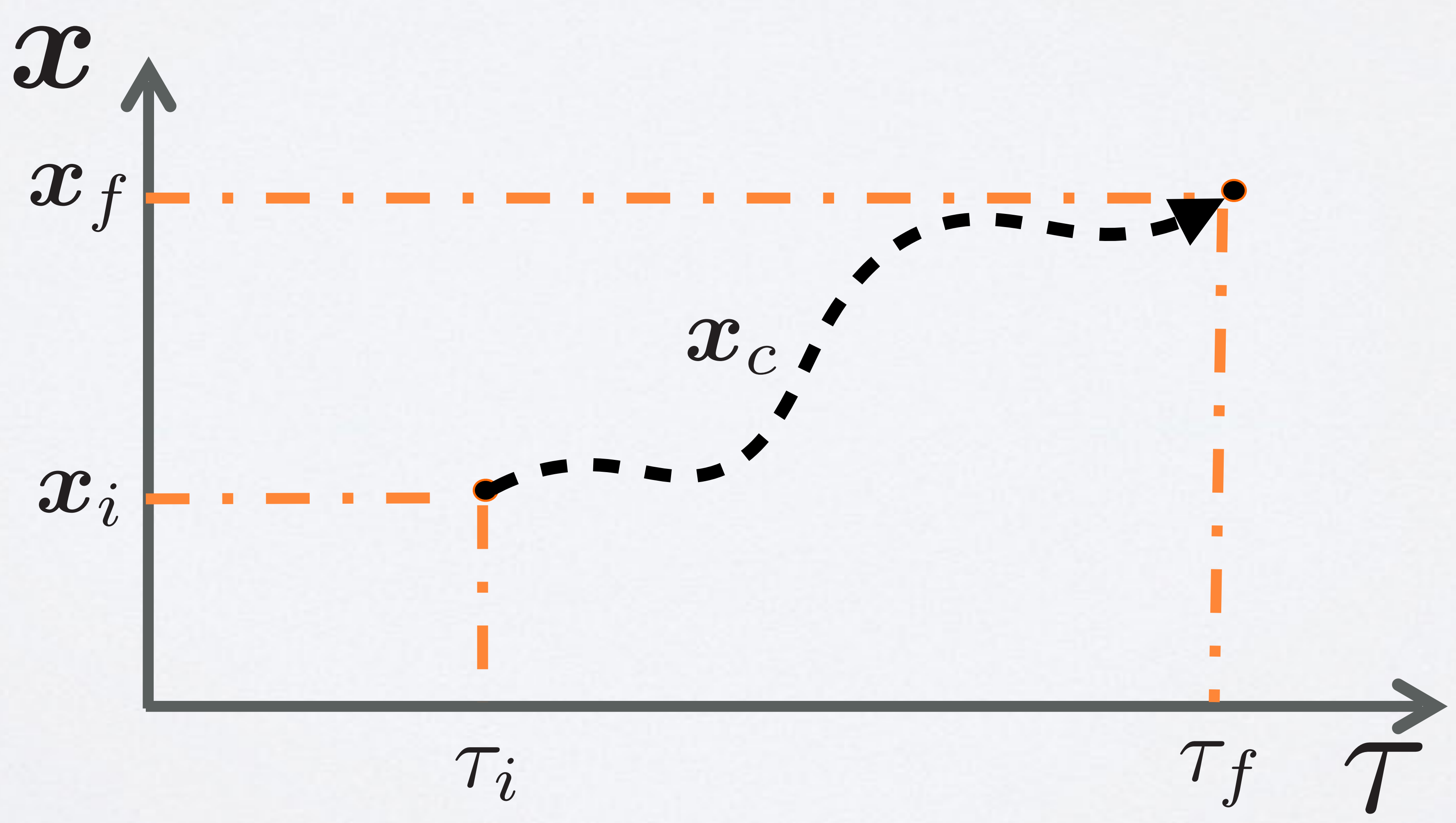


A FEW THINGS TO NOTE ABOUT THE EUCLIDEAN PATH INTEGRAL

- The points x_f and x_i do not have to be distinct. In principle, lots can be learned by setting $x_f = x_i$

- Clearly not all paths, $x(\tau)$, are created equal.
  - Each path is weighted by $\exp\left(-S[\boldsymbol{x}(\tau)]\right)$

- The "classical" path is defined where $\left.\frac{\partial S[\boldsymbol{x}(\tau)]}{\partial \boldsymbol{x}(\tau)}\right|_{\boldsymbol{x}_c(\tau)} = 0$



Problem #1 (simple):

Assume ***V(x,t)*** is the ***time-independent***, one-dimensional harmonic oscillator potential:

i.e.

$$V(x,t) = \frac{m\omega^2}{2} x^2$$

What is ***U(t',t)***?

What is the general solution for ***U(t',t)*** in the case of time-independent potentials?

Problem #2 (simple):

Assuming a ***time-independent*** Hamiltonian, show that the long-time behavior (i.e. t'>>t) of the evolution operator is

$$\lim_{\tau\to\infty} \langle x_0|U(\tau,0)|x_0\rangle \to |\Psi_0(x)|^2 \, e^{-E_0\tau}$$

where $\boldsymbol{E_0}$ is the system's ground state energy and $\boldsymbol{\Psi}_0(x)$ is the ground state wavefunction

Problem #3 (simple):

Assume

$$V(x) = \frac{m\omega^2}{2}x^2 + \lambda m^2\omega^3 x^4$$

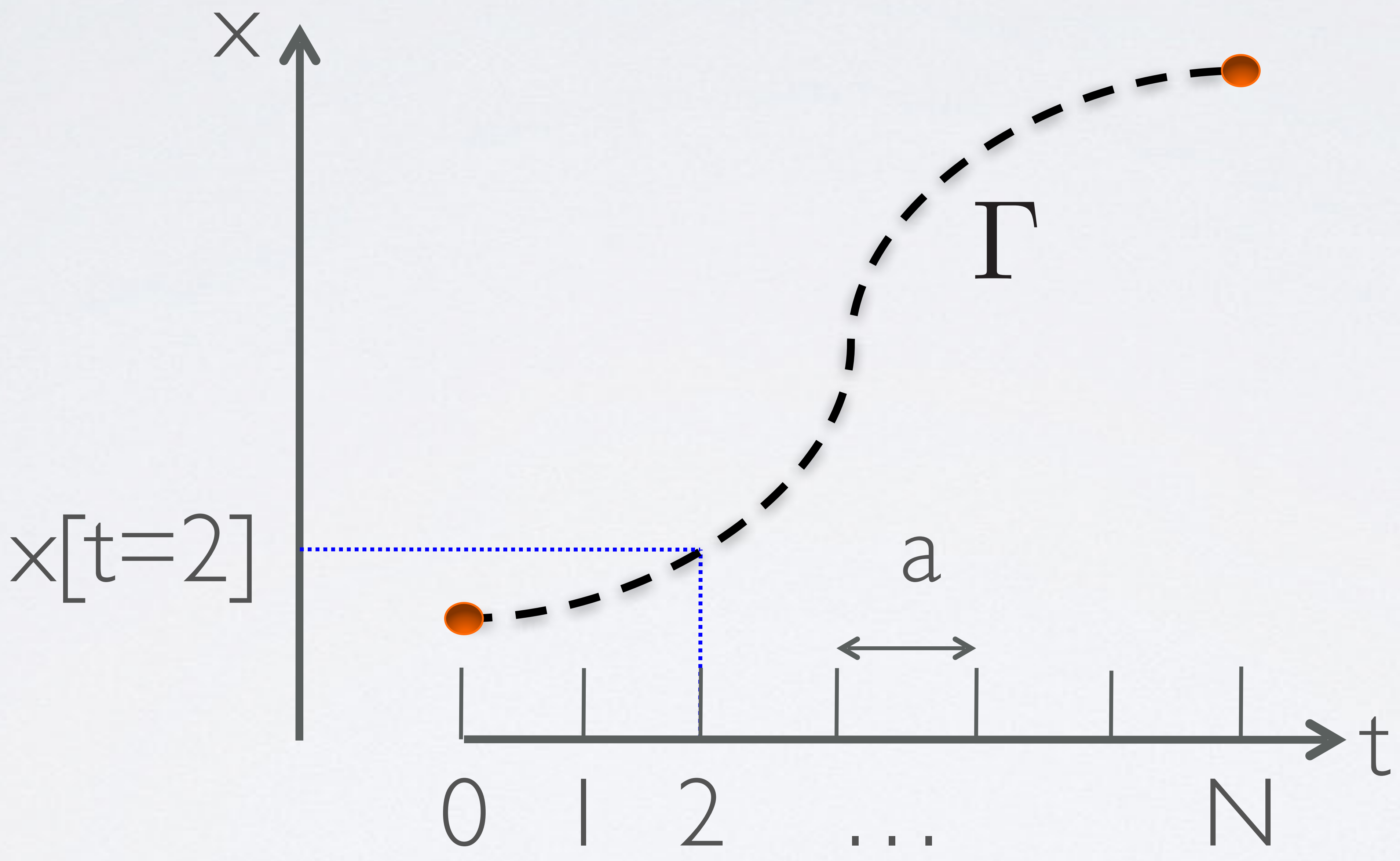
Use standard Rayleigh-Taylor perturbation theory to determine the ground-state energy shift to order **λ** and **λ**^{2}.

SO WHY IS THIS USEFUL?

- The Path-Integral formalism is amenable to numerics
  - It’s rather straightforward to put this formalism on a computer
  - We just need to “discretize” the formalism
- Not limited to “perturbative” interactions—should be able to do it all—well, in principle…

SO LET'S DISCRETIZE OUR 1-D HO EXAMPLE

- First: Let's discretize the time direction:



- Each path is represented by an array of position points
  - For example, the path above can be written as

$$\Gamma = \{x[0], x[1], x[2], \ldots, x[N-1], x[N]\}$$

AND OF COURSE, S[X] MUST NOW BE APPROXIMATED

$$\int_{ja}^{(j+1)a} dt \left(\frac{\dot{x}^2}{2m} + V(x) \right) \approx a \left(\frac{1}{2m} \left\{ \frac{x[j+1] - x[j]}{a} \right\}^2 + \frac{1}{2} \left\{ V(x[j+1]) + V(x[j]) \right\} \right)$$

$$S[x(t)] = \int_{t_i}^{t_f} dt \left(\frac{\dot{x}(t)^2}{2m} + V(x(t)) \right)$$

$$\Rightarrow S_{lat}[\Gamma] \approx \sum_{j=0}^{N-1} \left(\frac{m}{2a} \left\{ x[j+1] - x[j] \right\}^2 + \frac{1}{2} \left\{ V(x[j+1]) + V(x[j]\right\} \right)$$

AND THE INTEGRATION MEASURE BECOMES LESS ABSTRACT

$$\int [dx] \rightarrow \left(\frac{m}{2\pi a}\right)^{N/2} \int_{-\infty}^{\infty} dx[1]\; dx[2]\; dx[3] \ldots dx[N-1]$$

We don't integrate over endpoints $x[0]$ and $x[N]$ since they are fixed

Therefore:

$$\langle x_f | e^{-H(t_f - t_i)} | x_i \rangle = \langle x_f | U(t_f, t_i) | x_i \rangle \approx \left(\frac{m}{2\pi a}\right)^{N/2} \int_{-\infty}^{\infty} dx[1]\; dx[2]\; dx[3] \ldots dx[N-1] e^{-S_{lat}[x]}$$

LET'S TAKE A CLOSER LOOK AT WHAT WE'VE DONE

$$\langle x_f | e^{-H(t_f - t_i)} | x_i \rangle \approx \left(\frac{m}{2\pi a}\right)^{N/2} \int_{-\infty}^{\infty} dx[1]\ dx[2]\ dx[3] \ldots dx[N-1] e^{-S_{lat}[x]}$$

- We've only discretized the time direction

- At each point in time, x[j] can take on any value from –Infinity to +Infinity

- So essentially we've taken the path integral (a rather abstract object) and reduced it to an (N-1)-dimensional integral (a numerical object that can be simulated on a computer)

IT'S STILL NOT AN EASY PROBLEM, EVEN IN 1-D

- For accurate solutions, one ideally wants N to be large

- A layman's attempt would be to generate an ensemble of paths, or "configurations", $\{x\}$ at random and compute

$$\langle x_f | e^{-H(t_f - t_i)} | x_i \rangle \approx V \left(\frac{m}{2\pi a} \right)^{N/2} \frac{1}{N_{cf}} \sum_{\{x\}} e^{-S_{\text{lat}}[x]}$$

$$= V \left(\frac{m}{2\pi a} \right)^{N/2} \left\langle e^{-S_{\text{lat}}[x]} \right\rangle$$

number of paths in ensemble

"Monte Carlo" integration

volume of N-1 dimensional space

average value within ensemble

Problem #4 (simple):

Set $x_f = x_i = x$ and attempt the layman's approach to calculating the matrix element of the evolution operator. Try to extract the ground-state energy at large times.

Note: Sample points within a uniform distribution between -3.5 and 3.5, for example. In this case $V=(3.5)^{Nt-1}$, where Nt is the number of time slices.

Problem #5 (moderate):

Now add an interaction term

$$V_I(x) = \lambda m \omega^3 x^4$$

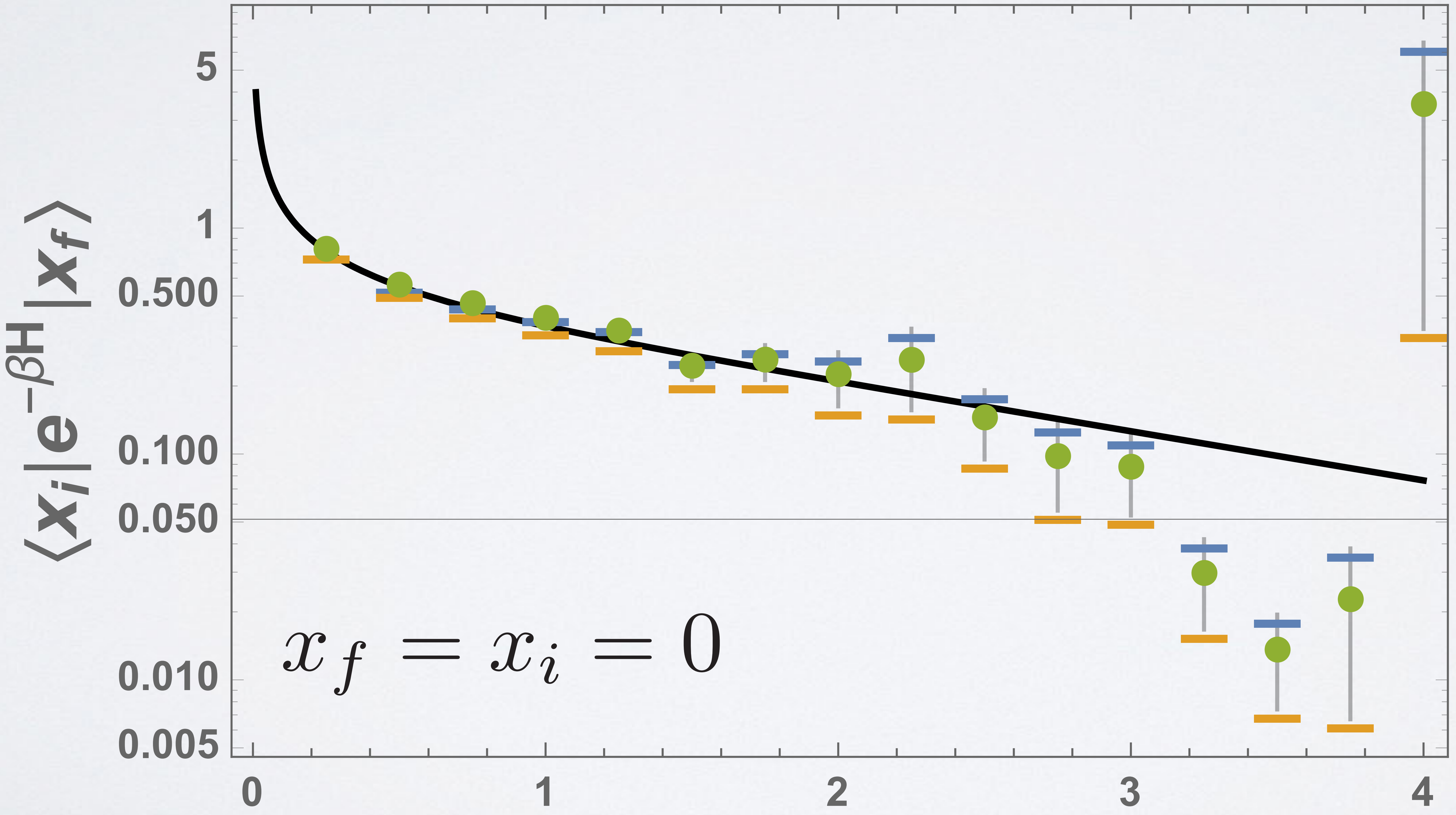
to the action. Investigate the behavior of the ground-state energy as a function of λ between 0 and 1. Overlay your results from Problem 4 to determine range of validity of your perturbative results.

LET'S LOOK AT PROBLEM 4

Turns out the evolution operator can be solved exactly for the 1-D HO

$$\langle \boldsymbol{x}_f, \tau' | \boldsymbol{x}_i, \tau \rangle = \langle \boldsymbol{x}_f | U_\omega(\tau', \tau) | \boldsymbol{x}_i \rangle = \frac{\exp\left(-\frac{\left(e^{-2\omega\beta}+1\right)\left(x_f^2+x_i^2\right)-4e^{-\omega\beta}x_f x_i}{2\left(1-e^{-2\omega\beta}\right)} - \frac{\omega\beta}{2} \right)}{\sqrt{\pi}\sqrt{1-e^{-2\omega\beta}}}$$

$$\beta = \tau' - \tau$$



$\omega\beta$ ⟶ "inverse temperature"

WHAT HAPPENS IF YOU HAVE TWO OR MORE (INTERACTING) PARTICLES?

- First off, if the particles aren't interacting, then the problem reduces to a one-body problem

Problem #6 (easy):

Show that for N non-interacting particles, the full path integral reduces to the product of N single-particle path integrals

- If the particles are interacting, then for two particles have, for example,

$$\langle \boldsymbol{x}_f \boldsymbol{y}_f | U(t', t) | \boldsymbol{x}_i \boldsymbol{y}_i \rangle = \int_{x(t)=\boldsymbol{x}_i}^{x(t')=\boldsymbol{x}_f} [d\boldsymbol{x}(\tau)] \int_{y(t)=\boldsymbol{y}_i}^{y(t')=\boldsymbol{y}_f} [d\boldsymbol{y}(\tau)]\, e^{-S[\boldsymbol{x}(t),\boldsymbol{y}(t)]}$$

$$S[\boldsymbol{x}(t), \boldsymbol{y}(t)] = \int_t^{t'} d\tau \left(\frac{\dot{\boldsymbol{x}}(\tau)^2}{2m} + \frac{\dot{\boldsymbol{y}}(\tau)^2}{2m} + V\left(\boldsymbol{x}(\tau), \boldsymbol{y}(\tau), \tau\right) \right)$$

EXPECTATION VALUES OF OTHER OPERATORS ARE EASY TO CALCULATE

- Given an operator ***O(x)***, the expectation value can be calculated as

$$\langle E_0|\hat{O}(\hat{x})|E_0\rangle = \frac{\int [d\boldsymbol{x}(t)]\, O(x) e^{-S[\boldsymbol{x}(t)]}}{\int [d\boldsymbol{x}(t)]\, e^{-S[\boldsymbol{x}(t)]}}$$

Problem #7 (moderate):

Prove it!

- The expectation value of ***O(x)*** is just the sum over all paths ***weighted by*** **exp(-S[x])**.

- We could just apply our layman's approach to this problem…

BUT WE CAN DO BETTER THAN THAT!

- The problem with the layman's attempt is that one spends lots of time generating configurations that are not relevant! In other words, the phase space being probed is too large.

- What we want is to generate configurations in such a way that the probability $P[x_n]$ of obtaining a particular configuration $\boldsymbol{x}$ is

$$\mathbb{P}\left[\boldsymbol{x}(\tau)\right] \propto \exp\left(-S[\boldsymbol{x}(\tau)]\right)$$

- This ensures that the generated ensemble of configurations have the highest probability of being relevant.

SO HERE'S SOME PSEUDO-CODE THAT DOES JUST THAT

Procedure to generate x_{n+1} given x_n:

Loop through x_n[j]

At site j, generate a random number χ uniformly distributed from –μ to +μ

Replace $x_n[j] \rightarrow x_n[j]+\chi$ and compute the change in action ΔS

If ΔS < 0, accept the new value of x_n[j] and continue to site j+1

If ΔS > 0, sample another number ρ uniformly distributed from 0 to 1. If exp(-ΔS) >ρ accept the new value of x_n[j], otherwise reject change. Continue to site j+1

```
def update(x, a, mu, N):
    global num_of_updates, num_of_accepts
    for j in xrange(1,N):    # we do not
        num_of_updates += 1 # change endpoints
        old_x = x[j]

        # now update x[j]
        x[j] = x[j]+uniform(-mu,mu)

        # this is the change in action
        dS = actions.deltaS_HO(j,x,old_x,a,N)

        # do we accept or not?
        if dS > 0 and exp(-dS) < uniform(0,1):
            x[j] = old_x   # don't accept change
        else:
            num_of_accepts += 1   # tally acceptance
            actions.actionS += dS # update action
```

Metropolis-Hastings

SOME POINTS TO CONSIDER

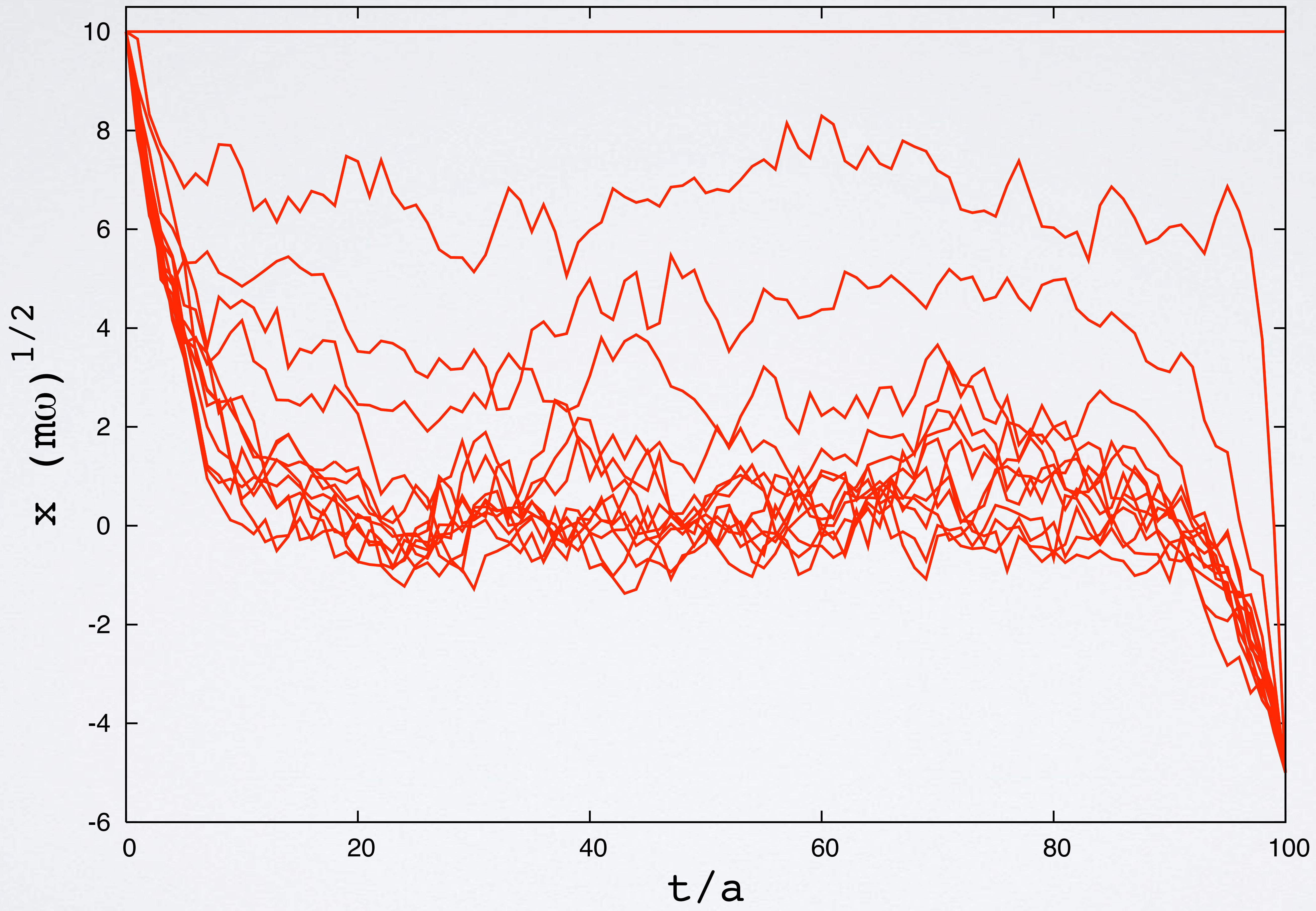
- This is the simplest example of the Metropolis Algorithm (they can get much more complicated)

- One has to start from some initial configuration x_1
  - Usually these initial configurations don't represent 'good' configurations
  - Run the algorithm for the first 100-1000 'trajectories', allowing the configurations to 'thermalize'—keep configurations afterwards

- One tunes **μ** such that one gets approximately ~70% acceptance rate

- In general configuration x_{n+1} is correlated to some degree with x_n—there are statistical methods to reduce these effects (e.g. binning, blocking, . . .)

SO HERE'S AN EXAMPLE OF THERMALIZING A CONFIGURATION



LET'S COME BACK TO OUR PROBLEM OF EXPECTATION VALUES

- Our original problem involved a sum over paths weighted by ***exp(-S[x])***

$$\langle E_0|\hat{O}(\hat{x})|E_0\rangle = \frac{\int [d\boldsymbol{x}(t)]\, O(x)e^{-S[\boldsymbol{x}(t)]}}{\int [d\boldsymbol{x}(t)]\, e^{-S[\boldsymbol{x}(t)]}}$$

- We've now generated an ensemble of paths {χ} with probability distribution

$$\mathbb{P}\left[\boldsymbol{x}(\tau)\right] \propto \exp\left(-S[\boldsymbol{x}(\tau)]\right)$$

- Our problem now turns into an ***unweighted*** sum over paths in our distribution

$$\langle E_0|\hat{O}(\hat{x})|E_0\rangle = \frac{1}{N_{cf}} \sum_{\boldsymbol{x}_i \in \{\boldsymbol{x}\}} O[\boldsymbol{x}_i] + \mathcal{O}(N_{cf}^{-1/2})$$

MY PYTHON CODES

- In each folder, there are various python routines
    - actions.py
    - metropolis.py
    - xxxx.py

Contains various function definitions— needs to be in the same directory with xxxx.py

Python executable

- To run any of the python executables, just type

```
>> python3 xxxx.py
```