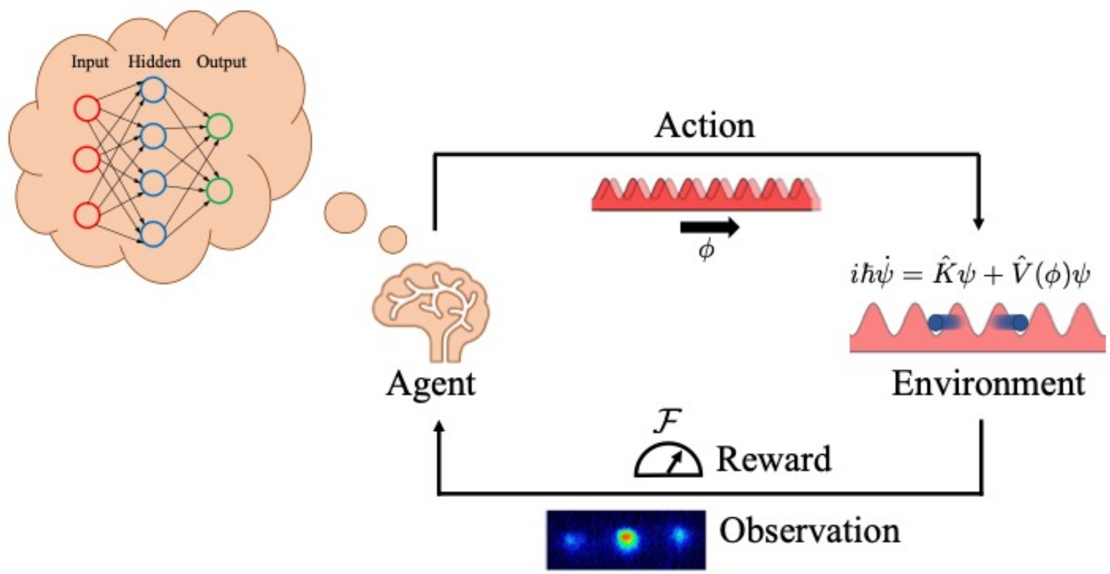
Using Machine Learning for the Quantum Design of a Matter-Wave Interferometer

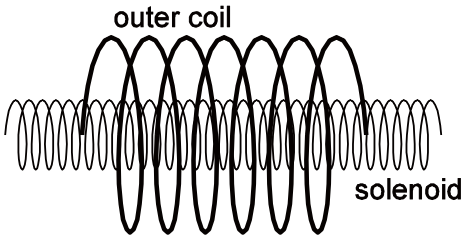


Murray Holland
JILA & University of Colorado Boulder

Atomtronics workshop, Benasque 2022

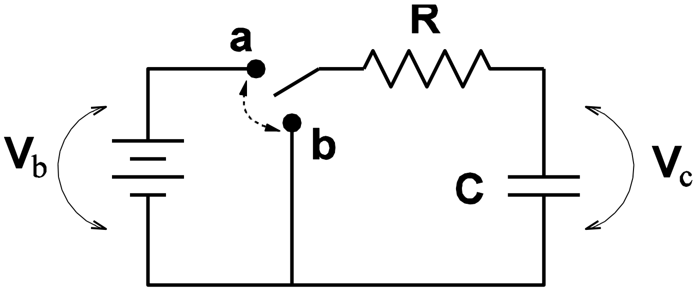
Acknowledgements: Dana Anderson, Liang-Ying Chih

Atomtronics



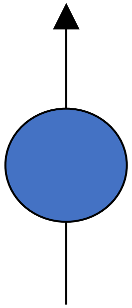
Electron systems two fundamental degrees of freedom

- Charge (flux of charge defines a current)
- Spin ½ (spintronics)



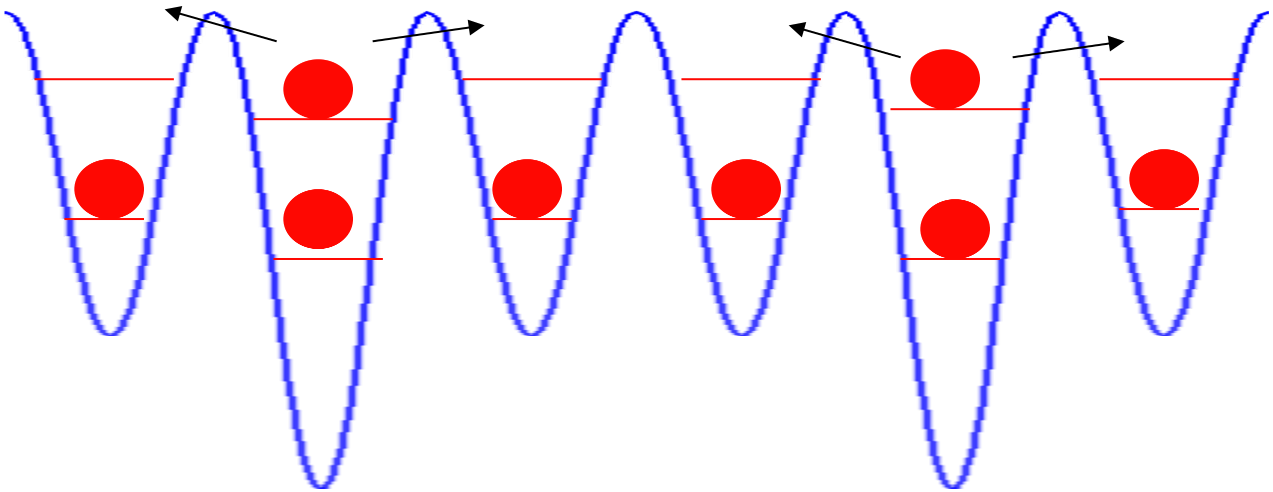
Atomtronics: analogy of semiconductor electronics in atomic systems

- Number (flux of particles defines a current)
- Spin-N (hyperfine quantum numbers)
- Boson / fermion, atom / molecule, …
- Coherence / Entanglement / Interference
- Superfluidity



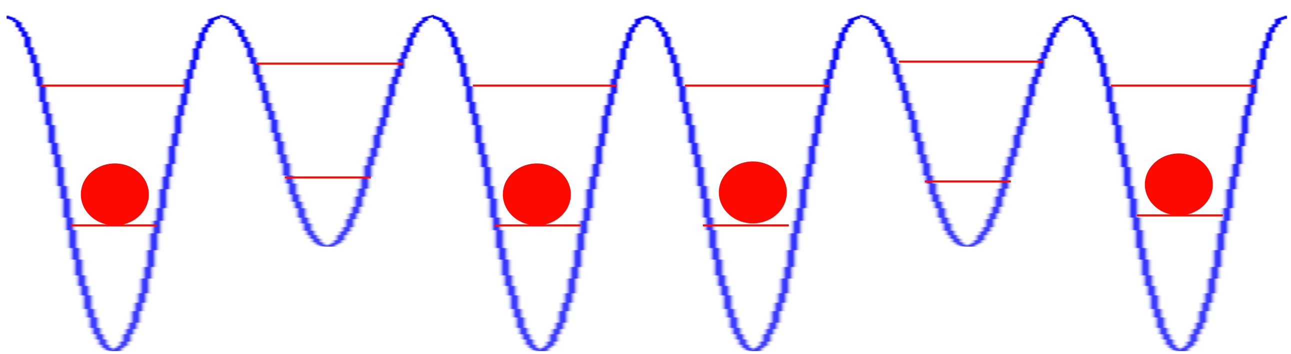
N-type doping

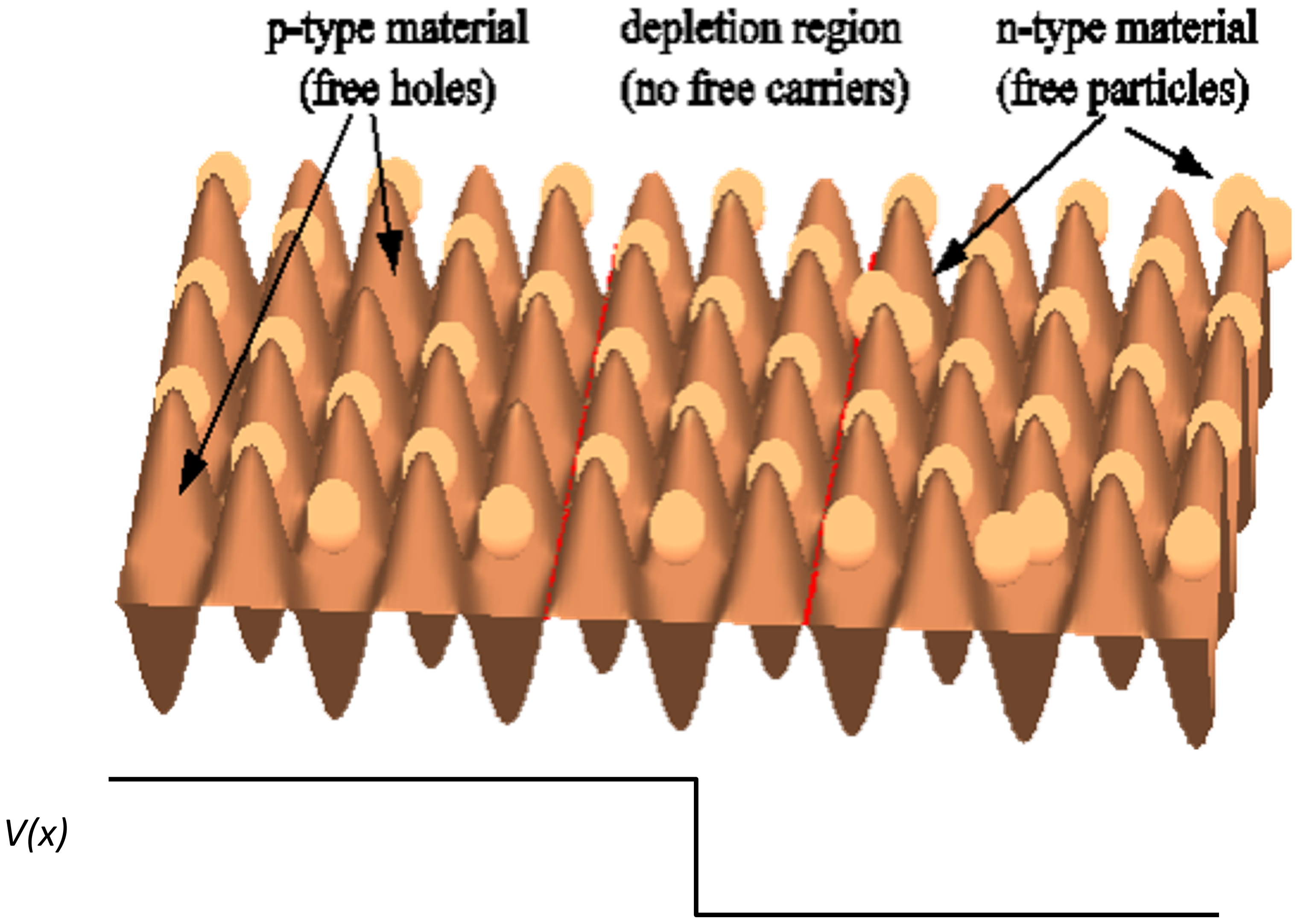
- extra atoms free to move



P-type doping

- extra holes free to move





Particle-hole recombination

Atomtronics: Quantum2.0

Typically basic constituents;

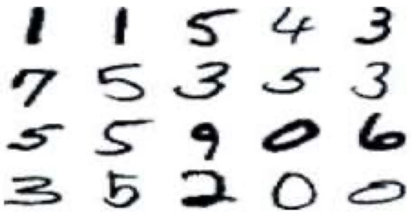
- Pristine building blocks
  (e.g., BECs, 2-level atoms)
- Simple potential surfaces
- One (or a few) electromagnetic modes

Quantum design of atomtronic circuits is complex;

- Large complex unitary 2^n
- Multiple paths / interference

Use machine learning to aid design?

Types of Machine Learning



- **Supervised Learning**: relation between the features/labels from *labeled data*
- **Unsupervised Learning**: underlying patterns from *unlabeled data*
- **Reinforcement Learning**: learn sequential decision making through trial and error

- Supervised/Unsupervised Learning: teach by examples
- Reinforcement Learning: teach by experience

- There are too many possible strategies so that it is impossible to do a brute-force search
- Human strategies are limited by our experience and imagination
- Many quantum system tasks fit this paradigm: Theory, Experimental Modelling, Data-driven

Example of machine learning in complex systems

Teaching robots to walk

- States: body position, terrain
- Actions: joint positions, angles
- Reward: 1 if it takes one step further

Learning to play chess, GO

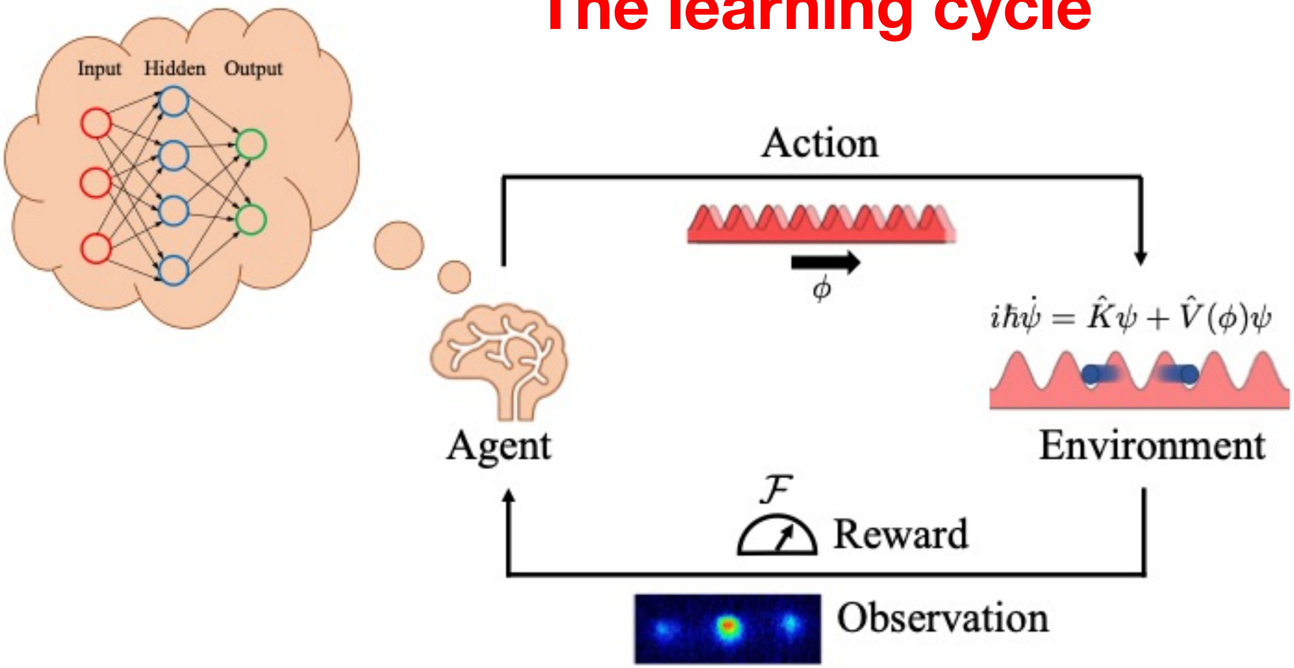
- States: board configurations
- Actions: where to place the pieces
- Reward: 1 = win, 0 = loss


This is Google's DeepMind AI
teaching itself how to walk
TECH
INSIDER

Reinforcement Learning Framework

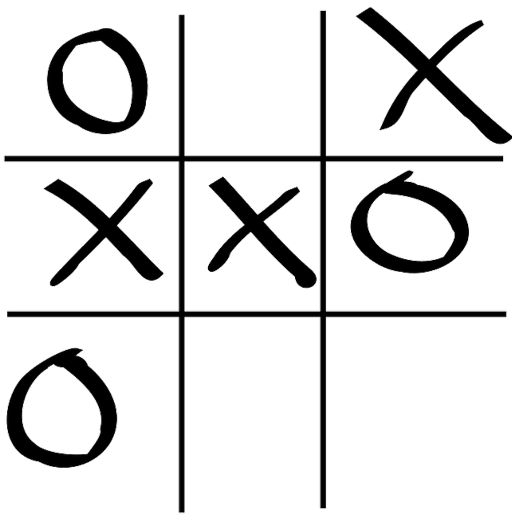
- **Agent**: which action?
- **Environment**: respond to the action, return the next state and reward
- **Goal**: find a strategy that maximizes the long-term rewards

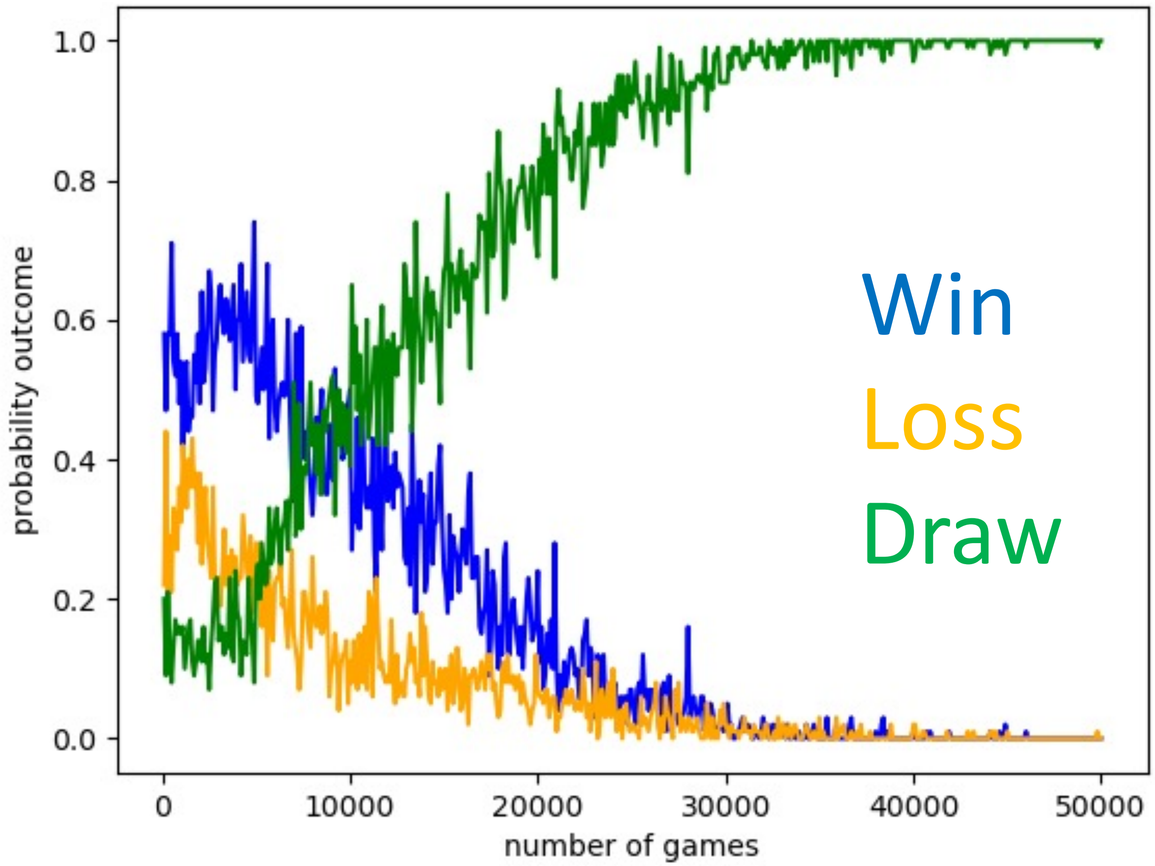
The learning cycle



Wish to apply this framework to the quantum engineering of designer platforms for carrying out specific tasks

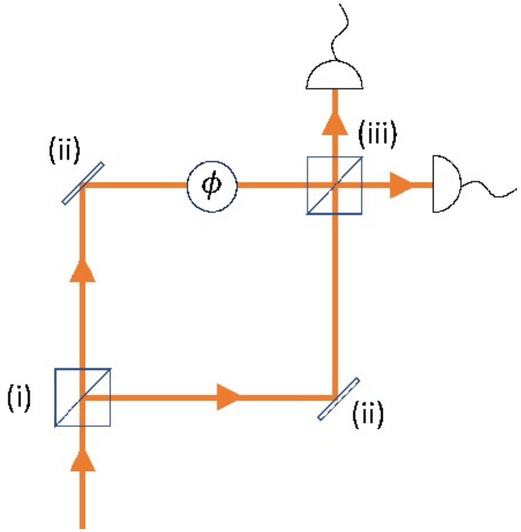
Tic-tac-toe



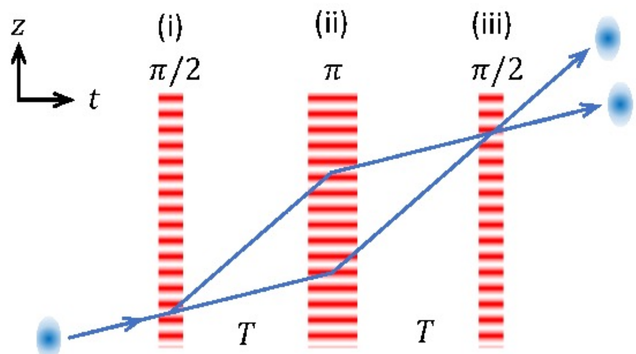


Model free learning: the agent doesn’t know if it is playing tic-tac-toe, chess, doing quantum design, or controlling an AMO experiment!

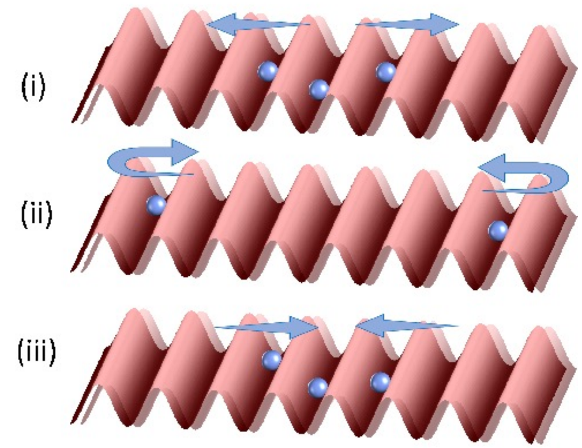
Machine learning for interferometry



Mach-Zehnder



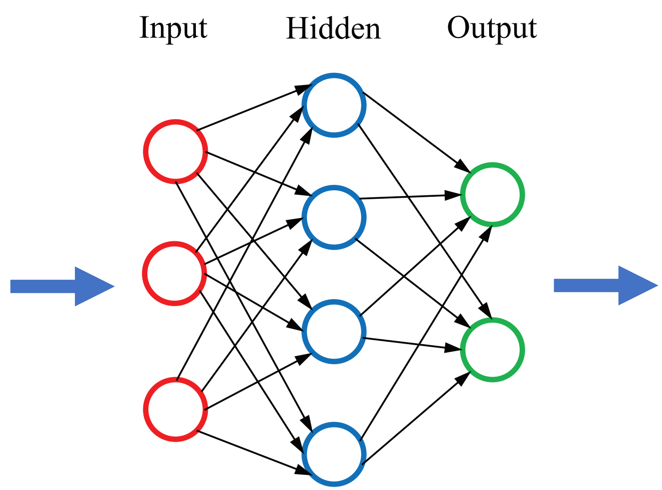
Bragg Interferometer



Shaken Lattice

Representation of the agent

Reduced Representation
e.g., populations



Action Values
e.g., lattice position

Agent

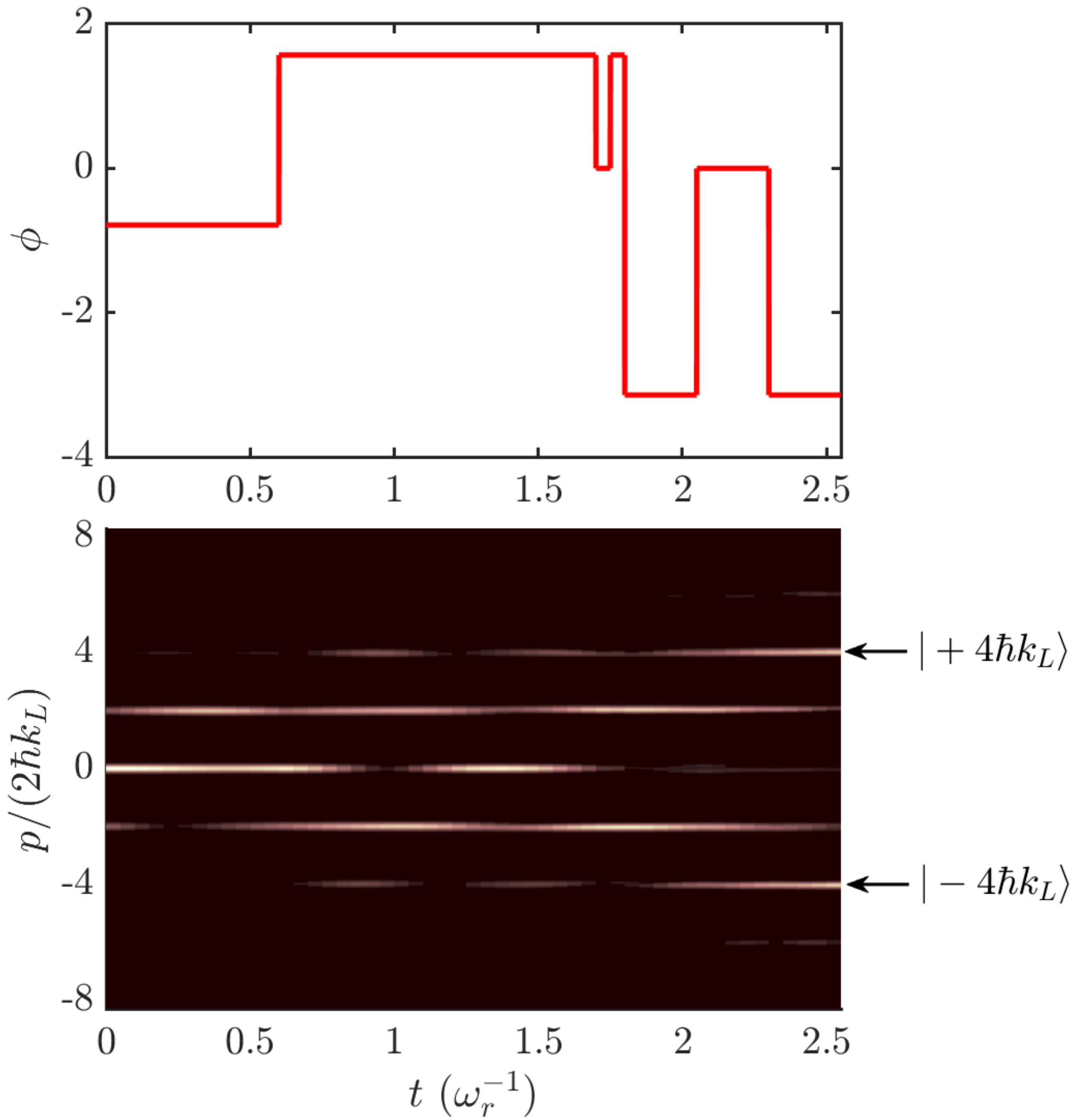
- Agent implementation is not unique
- Simplest is Tabular-Q: table of quality values for actions
- Can also use simple hidden-layer neural network
- More complicated strategies: Deep reinforcement learning (Double-deep Q networks = What we typically use)

Design philosophy

Design components and cascade
- Beam splitters
- Mirrors (i.e., reflectors)
- Free propagation
- Relative accelerators

Cost / Reward based on:
- Minimal detectable phase shift
- Dynamic range
- Total time taken
- Quantum/Classical Fisher information:
  Variation of output with control parameter

Beam Splitter

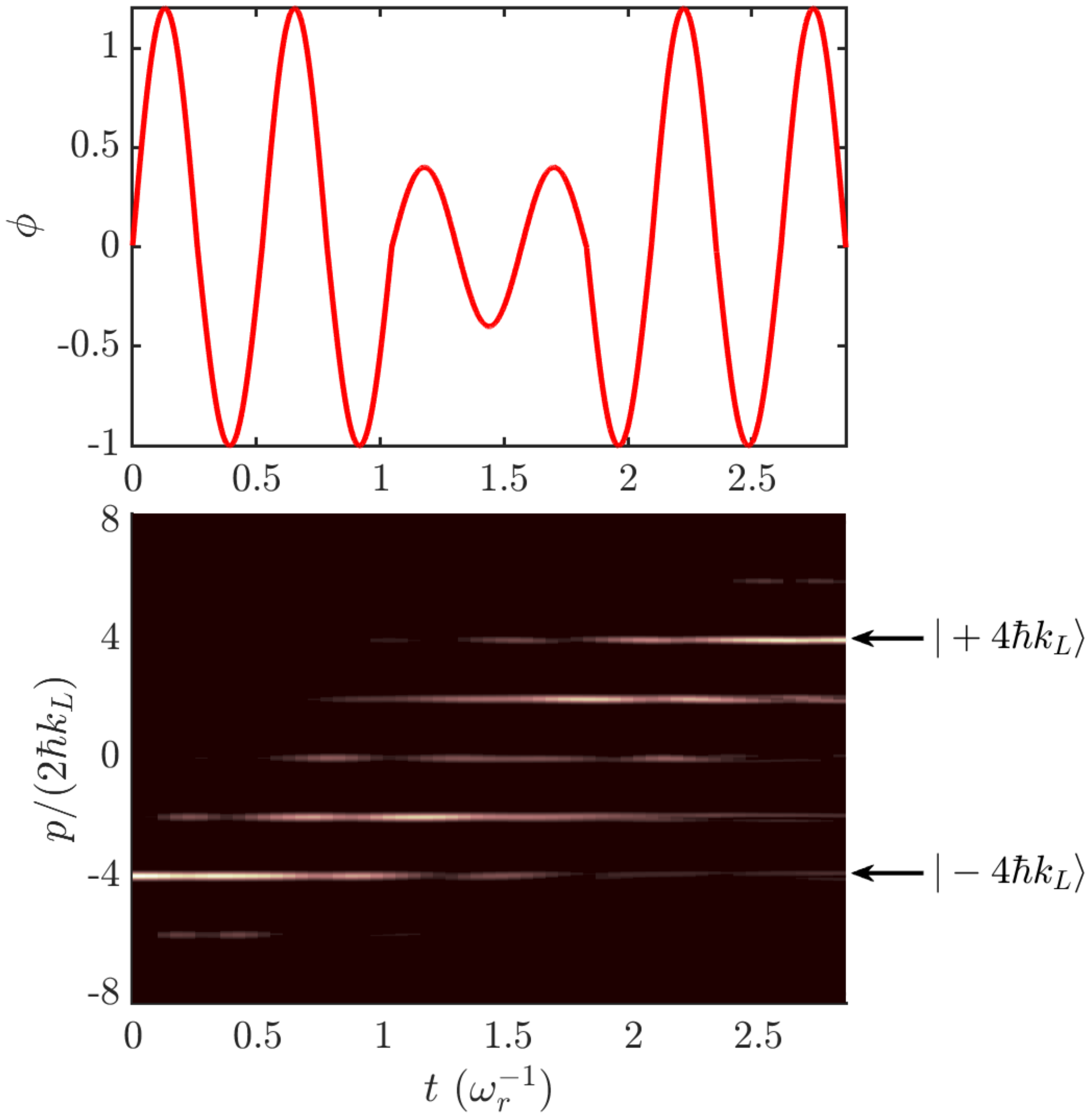


Target State

Reduced Rep: Momentum Prob

$|\Psi_i\rangle \rightarrow |\Psi_f\rangle$

Mirror



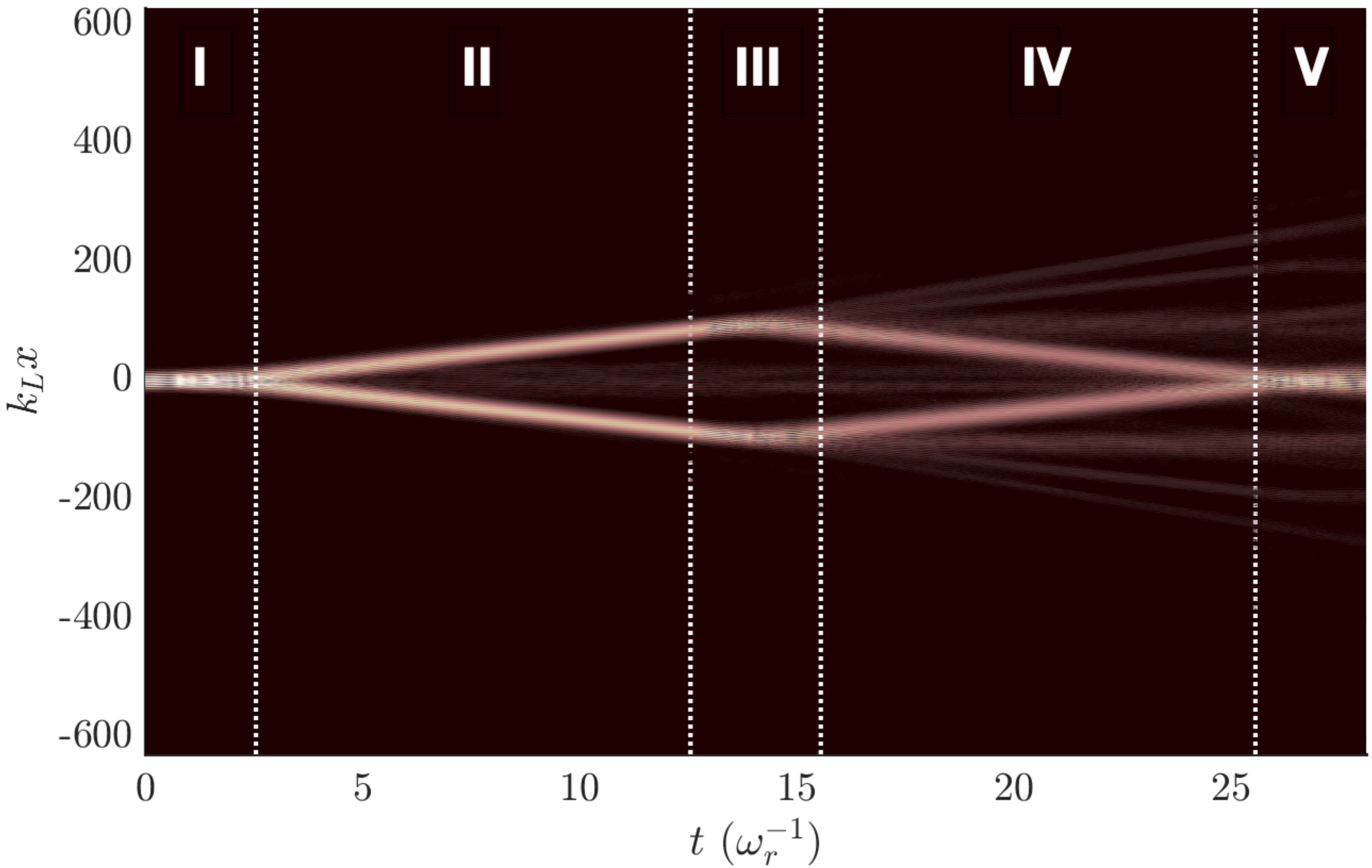
Target Operator

Reduced Rep:

Unitary Diagonal

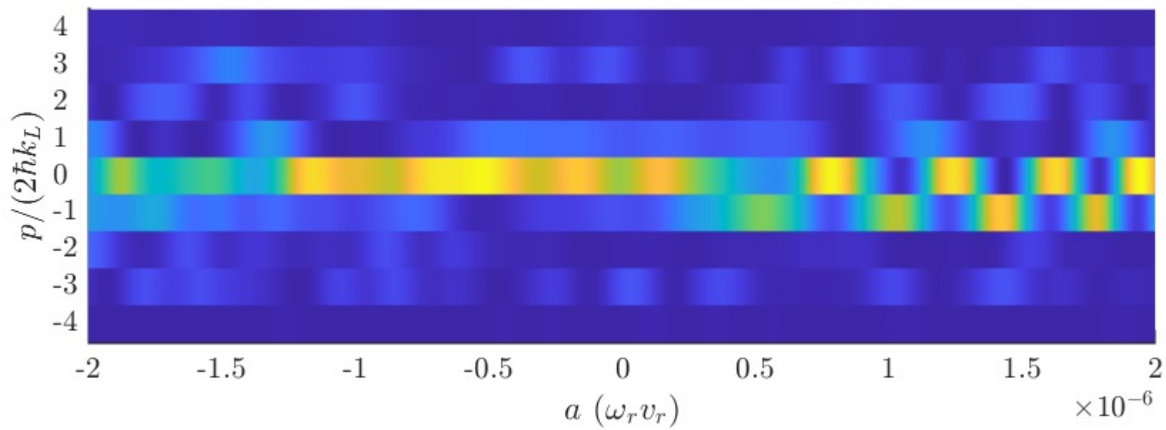
$\boldsymbol{U}_{\mathbf{mirror}}$

Real-space interferometer



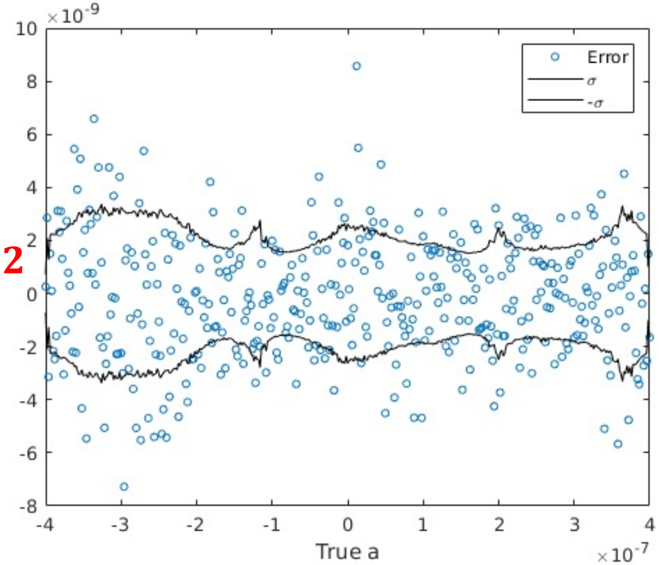
Response to acceleration signals

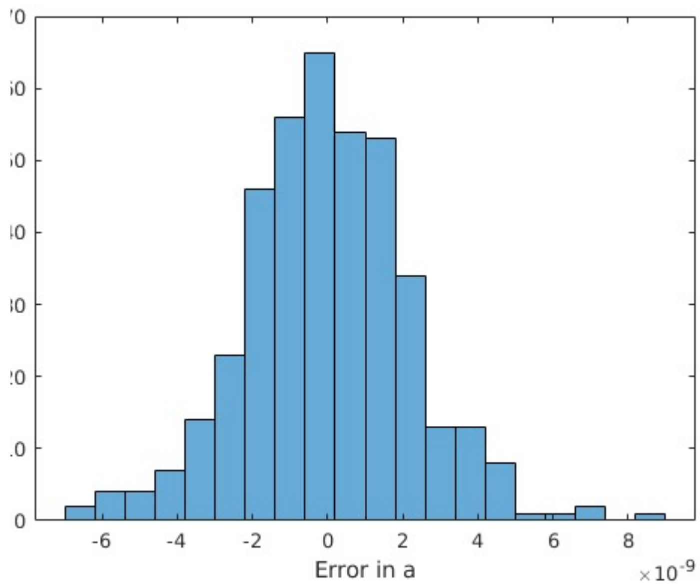
Each vertical slice: probability distribution of momentum for a certain value of acceleration



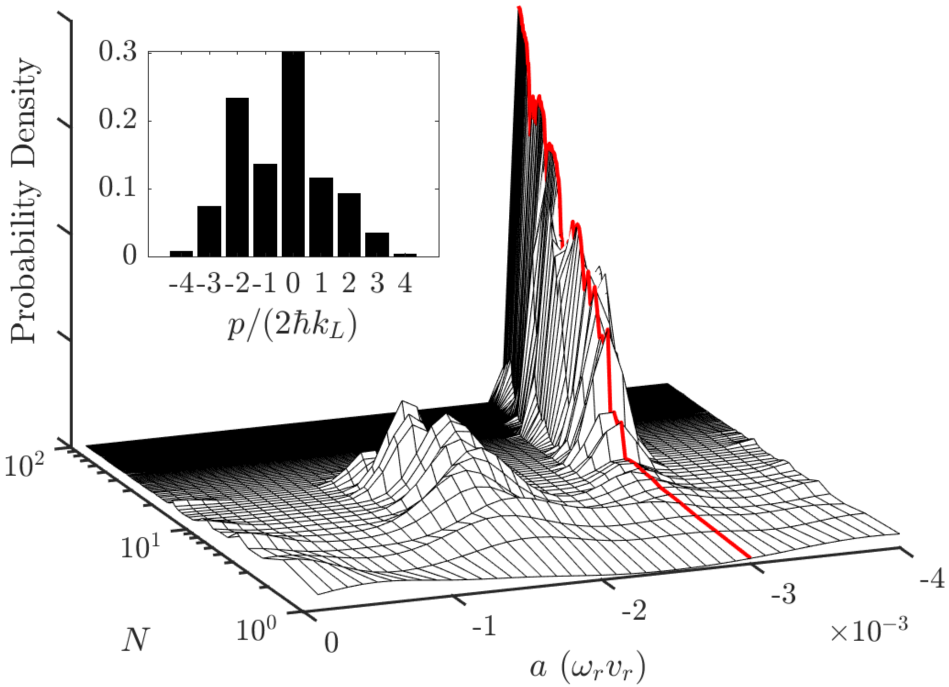
Fingerprint: no aliasing

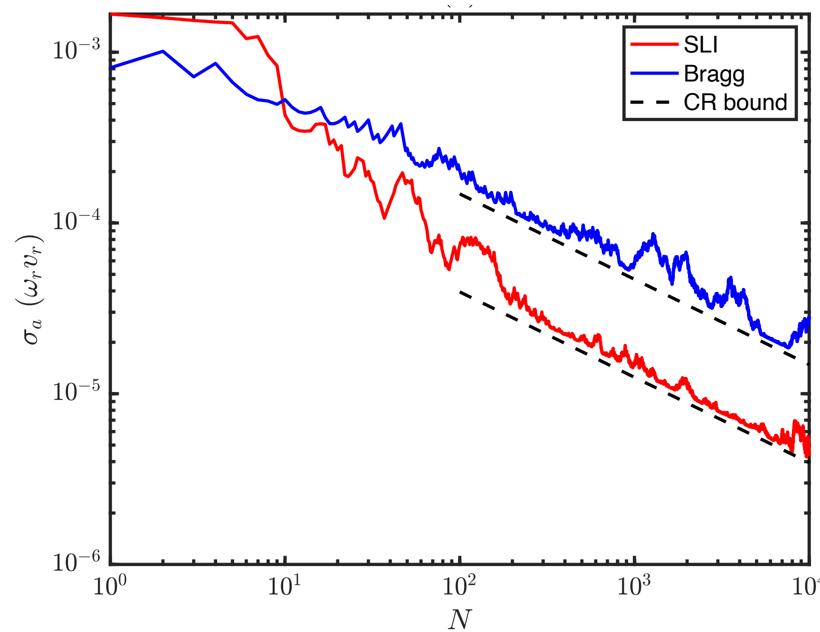
$\approx \mathbf{10^{-9}\ ms^{-2}}$





Parameter Estimation

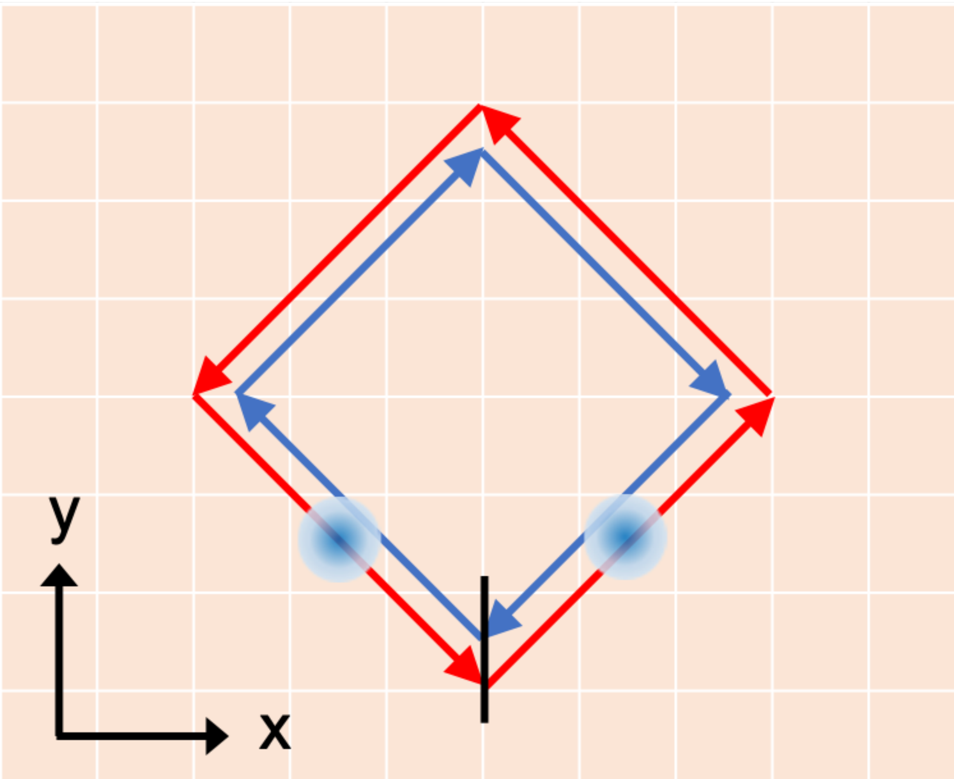




Repeated measurements
Bayesian Inference

Sensitivity
Standard deviation

Reinforcement learned gyroscope



- 1D beam-splitting protocol in x
- Free propagation for T
- 1D Reflecting protocol
- Free propagation for T
- Invert y motion
- Free propagation for T
- 1D Reflecting protocol
- Free propagation for T
- 1D recombination protocol

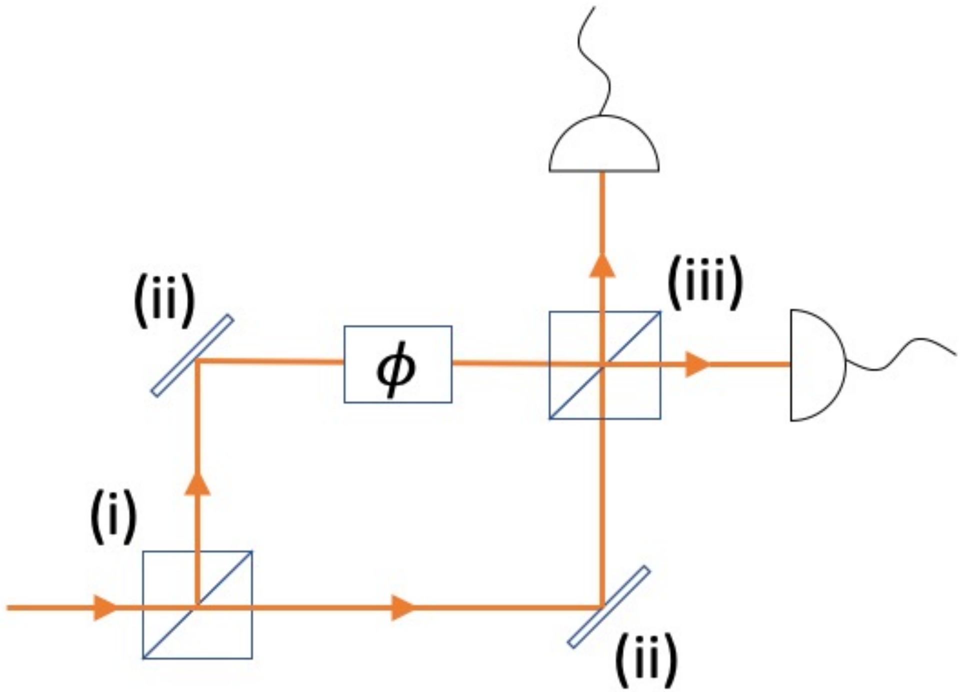
More difficult to model: two-dimensional environment

Separation ansatz for different dimensions; x, y, may work well

$$\boldsymbol{\psi(x, y) = \alpha(x)\beta(y)}$$

End-to-end design with reinforcement learning

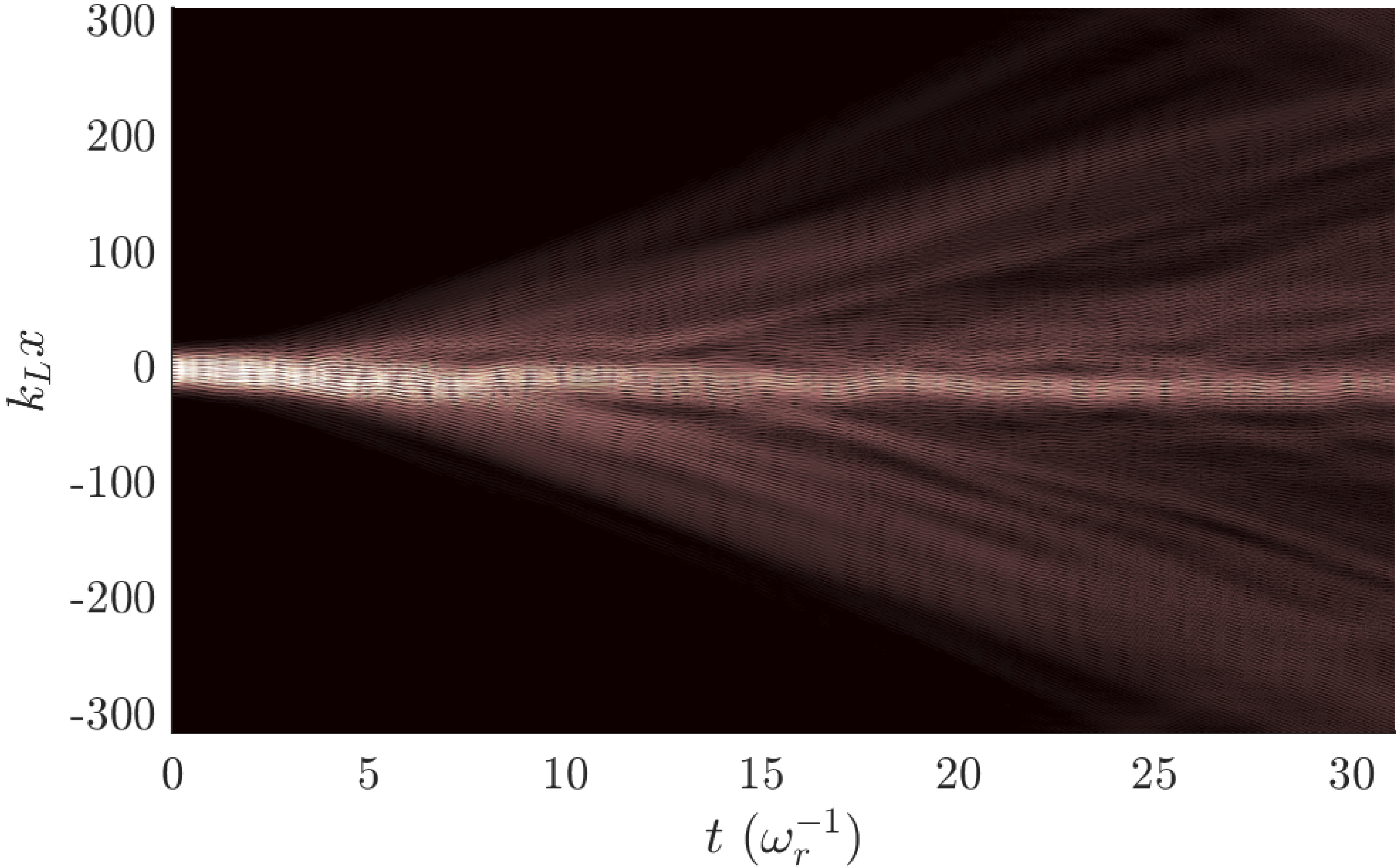
Why decompose into components?



Mach-Zehnder

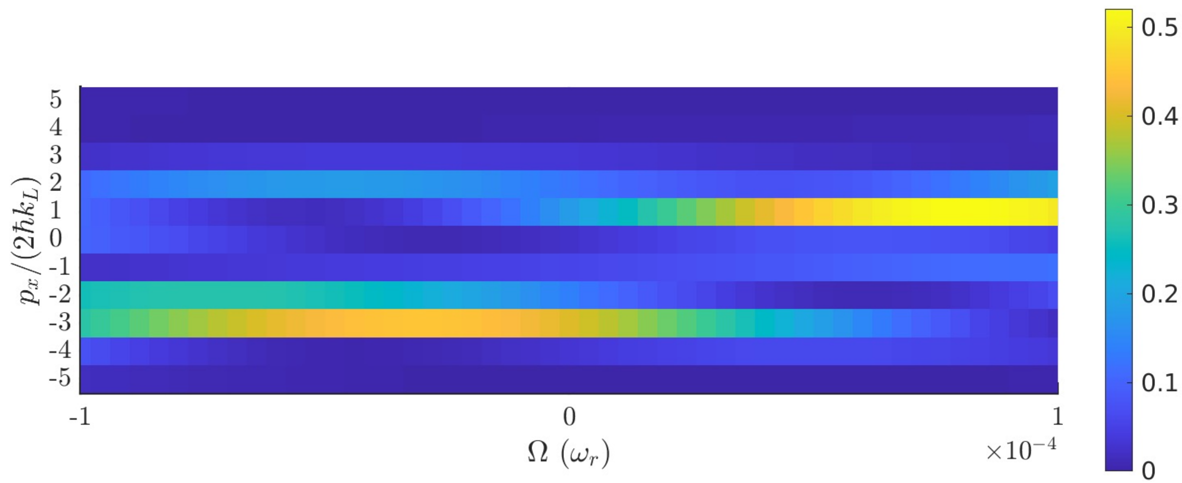
End-to-end design with reinforcement learning

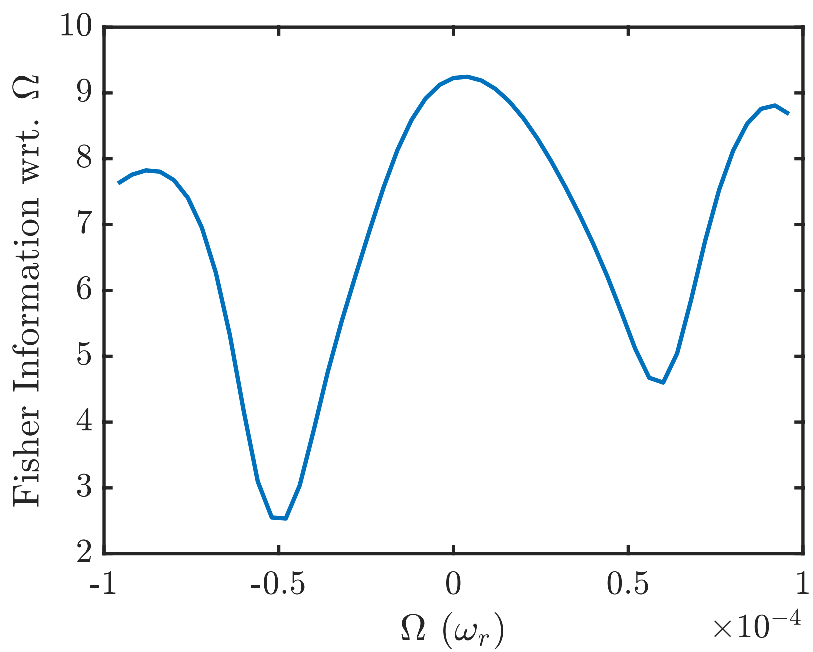
Why decompose into components?



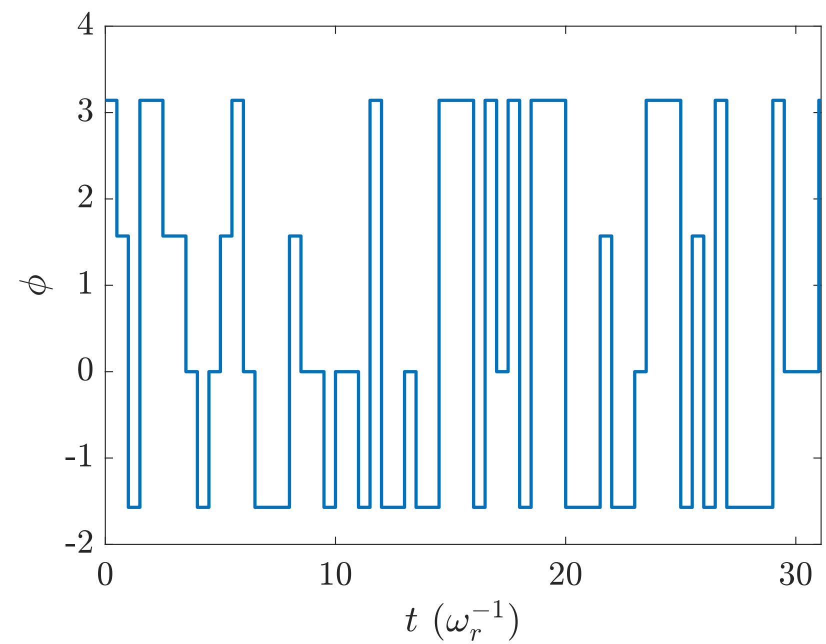
Like a multimode fiber interferometer, but not sensitive to strain/temperature etc.

Momentum distribution versus rotation of the gyro



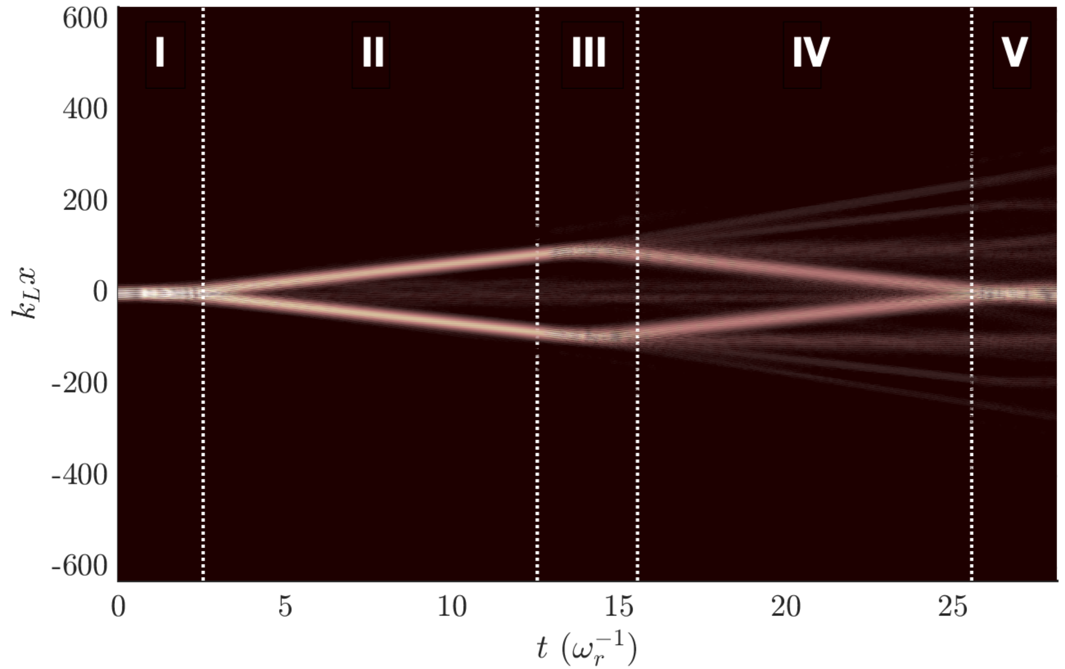


Fisher information relative to Bragg interferometer



Learned control-phase behavior

Summary



- Talked about machine learning as a general approach to design of atomtronic circuits
- Example matter-wave interferometer
- Access to non-intuitive solutions / components / devices

For more information; see for example:

"Using Machine Learning for the Quantum Design of a Matter-Wave Interferometer", Physical Review Research 3, 033279 (2021).