Reinforcement-learning-based matter-wave interferometer in a shaken optical lattice

Liang-Ying Chih and Murray Holland
JILA, NIST, and Department of Physics, University of Colorado, 440 UCB, Boulder, Colorado 80309, USA

(Received 21 June 2021; accepted 30 August 2021; published 27 September 2021)

We demonstrate the design of a matter-wave interferometer to measure acceleration in one dimension with high precision. The system we base this on consists of ultracold atoms in an optical lattice potential created by interfering laser beams. Our approach uses reinforcement learning, a branch of machine learning that generates the protocols needed to realize lattice-based analogs of optical components including a beam splitter, a mirror, and a recombiner. The performance of these components is evaluated by comparison with their optical analogs. The interferometer's sensitivity to acceleration is quantitatively evaluated using a Bayesian statistical approach. We find the sensitivity to surpass that of standard Bragg interferometry, demonstrating the future potential for this design methodology.

DOI: 10.1103/PhysRevResearch.3.033279

I. INTRODUCTION

Quantum metrology is an important field of quantum physics with the goal to make accurate and precise measurements of important physical quantities, ideally at a level that surpasses that achievable by any classical approach. In particular, extensive efforts have been made to develop metrological devices based on interference and detection of either electromagnetic waves or matter waves. By exploiting the quantum aspects of superposition and entanglement, one can potentially achieve high sensitivity to phase shifts, and this has inspired a wide variety of applications, including detecting gravitational waves [1], measuring the fine-structure constant [2], testing the universality of free fall [3], and inertial sensing for GPS navigation [4,5].

The usual direct-design approaches to engineering complex quantum systems that consist of many degrees of freedom or many particles are typically founded on experience with simpler analogs or intuition for underlying mechanisms. This naturally leads to a paradigm that is most accessible in terms of understanding, but incorporates human bias that may potentially generate nonoptimal solutions. With this perspective in mind, we point out there are a few purely systematic methods that are often used as a way to develop unbiased strategies, including optimal control [6] and optimization algorithms such as the Nelder-Mead simplex [7] or simulated annealing [8]. Utilizing these methods can allow one to explore solutions with more complicated forms with the potential to reach closer-to-optimum control protocols.

Recently, it has been shown that quantum design may benefit tremendously from a branch of machine learning based on trial and error, known as reinforcement learning, that aims to employ machines to find the optimal strategy for accomplishing a specific task. One of the reasons for its success is that the learning framework is decoupled from human intuition and therefore may explore novel solutions that have been previously undiscovered. Moreover, in situations where problems are so complex or extensive that naïve brute-force algorithms are considered unfeasible, such as playing games like *go* and *chess* [9–11], sophisticated reinforcement-learning algorithms have been developed to enable the machines to perform at a level that exceeds human capability. There are many systems in quantum physics that fit naturally within this scope due to their underlying complexity. For example, reinforcement learning has been applied to control and study phase transitions of many-body quantum systems [12], to find strategies for quantum error correction [13,14], to design quantum circuits [15], to prepare novel quantum states [16–18], to find protocols for quantum communication [19], and to improve quantum sensors [20]. In this paper, we will investigate the potential for reinforcement learning to improve on existing matter-wave interferometery. In Sec. II, we describe the physical model for the interferometric system. We outline our reinforcement learning approach in Sec. III and explain the key concepts crucial to understanding its application. In Sec. IV, we demonstrate how the reinforcement-learning framework may be used to design a beam splitter and a mirror and present the protocols that are learned. Following this we show how the components can be cascaded together to form a complete interferometer. In Sec. V, we evaluate the resulting performance of the interferometer through a Bayesian statistical analysis.

II. PHYSICAL MODEL

Although the topology of interferometers can vary somewhat, the archetypal design can be thought of as the Mach-Zehnder interferometer [21,22]. This device is sensitive to the differential phase accumulated between two alternate paths [see Fig. 1(a)]. Mach-Zehnder interferometers are composed from three essential components: A beam splitter that separates the wave coherently into two directions, mirrors

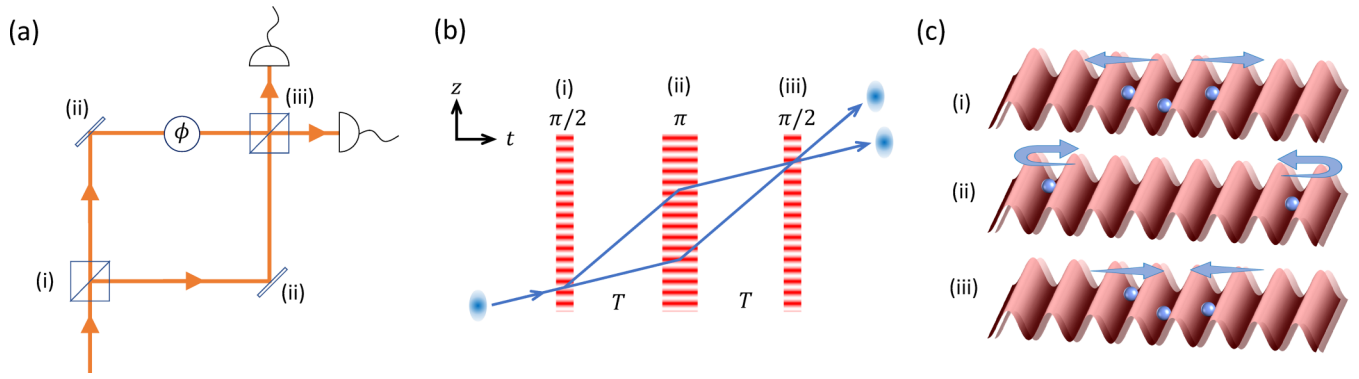

FIG. 1. Interferometers are composed of (i) a beam splitter, (ii) mirrors, and (iii) a recombiner. Examples shown are (a) an optical Mach-Zehnder interferometer (using half-silvered or conventional mirrors), (b) a Bragg interferometer (using three short light pulses of varying pulse area, $\pi/2$ or π), and (c) a shaken lattice interferometer. In (c), the shaken lattice mimics the interferometer components by splitting, reflecting, and recombining the atoms through design of a specific shaking function for each case.

that reflect both parts, and a recombiner where the waves are brought back together and constructively or destructively interfere. While in the optical case components that split or reflect light beams are readily available (e.g., half-silvered or conventional mirrors), for matter-wave optics, the components have to be generated through the careful control of laser-atom interactions. The traditional way to do this is through Bragg diffraction [see Fig. 1(b)] [23,24], in which a sequence of three short light pulses are used to separate, reflect, and recombine the matter wave.

An alternative approach is possible for atoms moving in an optical lattice potential where the intensity pattern can be shifted backward and forward in time [25–27]. The elementary components, i.e., a beam splitter, reflectors, and a recombiner, in that case may be implemented by "shaking" the lattice in a tailored pattern. If we denote the canonical position and momentum operators of an atom by $\hat{x}$ and $\hat{p}$, respectively, with m the atom mass, the optical lattice system can be described by the Hamiltonian

$$\hat{H}(t) = \frac{\hat{p}^2}{2m} - \frac{V_0}{2}\cos[2k_L\hat{x} + \phi(t)], \tag{1}$$

where k_L is the laser wave number, and $\phi(t)$ is the time-dependent phase difference between two counterpropagating lasers. The lattice, with constant amplitude V_0, is shaken through the variation of $\phi(t)$, since that parameter determines the position of the nodes and the anti-nodes of the standing wave intensity.

Unlike a typical Mach-Zehnder interferometer, which is formally a two-port device and transforms quantum states according to simple SU(2) group rotations, the relevant eigenstates of the shaken lattice potential consist of many Bloch states that can be coupled by the time dependence of the laser phase, $\phi(t)$. In some sense this situation represents a highly multipath form of interferometry since the Bloch basis provides many accessible paths for the quantum wave function to explore. While this establishes a rich evolution and could provide metrological benefit, it makes it more difficult to design an intuitive control protocol. It is for this goal of obtaining a high performance solution among a complex landscape that we are led to consider reinforcement learning as a design approach.

III. MODEL-FREE REINFORCEMENT LEARNING FOR DESIGN

In order to understand our design philosophy, we first need to address the important considerations to make when using machines to find solutions to design tasks. In this section we will present the principal ideas and concepts that will form the underlying foundations of our learning-based methodology. Since there are a variety of methods available, we begin with a discussion of the main structure that we will employ.

Reinforcement learning consists of a closed loop in which an *agent* invokes *actions* based on the observed *state* of an *environment*, and an environment provides *rewards* to the agent based on the observed outcome of its actions. The agent is tasked with the goal to discover the sequence of actions for which it receives the highest possible terminal reward [28] (see Fig. 2). It does this by trial and error, iteratively improving its actions in such a way as to corral the environment toward a target configuration.

We are primarily interested in a specific kind of reinforcement learning, known as model-free learning [28], where the

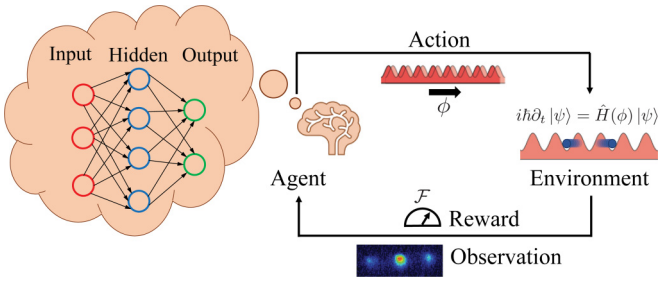

FIG. 2. Framework of reinforcement learning. The agent chooses an action, the environment responds to the action and gives the next state and a reward as feedback. In our design task, the action is the translation of the optical lattice, the environment evolves according to the Schrödinger equation, the observation is the momentum population distribution, and the reward is a function of the quantum fidelity. Illustrated on the left is an example neural network as the decision-making agent. The neural network takes the state represented on its input layer, passes it through a hidden layer, and generates a vector of Q-values at its output layer.

agent has no detailed insight as to the structure of the environment and cannot know *a priori* what effect an action will have on the environment state. Here the agent learns and makes decisions even when the environmental model may not be fully known. This general setting can even be applied to the situation in which the environment is represented by an experimental apparatus that cannot be fully understood, and the reward is derived from experimental observation and feedback. Using this approach significantly expands the potential scope of our work, since it is precisely this aspect that distinguishes model-free reinforcement learning from classical optimal control methods. In classical optimal control, the optimization depends heavily on the mathematical form of the model and, for complex systems, can be difficult to implement in practice.

Typically, reinforcement learning problems are Markov decision processes, where the probability of transition to the next state is only dependent on the current state and the current action and not on prior history. Evolution in this framework is referred to as a *trajectory*, where the state is initially prepared and then a sequence of actions and corresponding state updates are performed. The trajectory steps continue until the chain comes to an end when a predetermined condition is reached, which might be a certain number of steps, or a terminating state of the environment. An important concept is the idea of an *optimum trajectory* starting from any state, which is a trajectory such that the terminal reward is the maximum possible. We use what is known as Q-learning to make the action decisions, as we now describe. Although this approach is not the only possible choice, it is appropriate when the effect of an action on the environment state is deterministic, which will be the case here.

Since there is potentially an enormous dimensionality of the state space and the action space, a brute-force search is not possible, and we employ a neural network to learn and to approximately optimize the crucial decision-making task. That is, we define the input nodes of a neural network to be a representation of an arbitrary vector in the state space. We ascribe to the output of the neural network a vector of quality factors, simplified to Q-values, $Q(s, a) \in \mathbb{R}$, where each element represents the desirability for a possible action, a, given an input state, s. Ideally the neural network should take *any* arbitrary state as input and output a distribution in which the maximum Q-value corresponds to the best next action to perform.

We train the neural network in the following manner. The agent will utilize a policy in order to decide on the next action for a given state, for example, by taking the action associated with the maximum Q-value from the neural network output vector. The environment, in some initial state s, receives the action a from the agent and then updates the state, symbolically denoted as

$$s \xrightarrow{a} s'.$$

The environment then reports back to the agent the reward, $r(s')$. As a step in a sequence, this leads to the important concept of the discounted cumulative reward, known simply as the *return*. The return, $Y(s')$, is defined as the combination of a current reward, $r(s')$, and every future reward that would be generated in an optimum trajectory seeded by s'. Since any current action will have decreasing influence on future decision making as the steps become more distant, it is useful to deweight each consecutive reward by a discount factor, $\gamma \in [0, 1]$, i.e.,

$$Y(s') \equiv r(s') + \gamma r(s'') + \gamma^2 r(s''') + \ldots, \tag{2}$$

where

$$s' \xrightarrow{a'} s'' \xrightarrow{a''} s''' \xrightarrow{a'''} \ldots$$

are steps along an optimum trajectory.

The learning process is now framed as the minimization of the squared difference between the return, $Y(s')$, and the network output, $Q(s, a)$, where the minimization is accomplished by varying the internal weights and biases that constitute the neural network. Should the learning be perfected, the squared difference would be zero, implying that the Q-value and corresponding return would be identical. We emphasize that, while mathematically complete, this learning definition is a formal construction that is intractable as written, since if we knew the optimum trajectory there would be no need to carry out the learning at all. In fact, the only information that is accessible about the future component of $Y(s')$ is the inference one can make from the neural network output itself, introducing a self-consistency element to the optimization problem. In other words, the series in Eq. (2) is approximated as

$$r(s'') + \gamma r(s''') + \ldots \approx \max_{a'}[Q(s', a')], \tag{3}$$

so that

$$Y(s') = r(s') + \gamma \max_{a'}[Q(s', a')]. \tag{4}$$

The final result given in Eq. (4) encapsulates all the principal concepts that establish a closed learning formulation that can be computationally implemented.

We employ several technical but important improvements that increase the convergence, such as to replace the neural network used for the calculation of the future return, as approximated in Eq. (3), with a second identical neural network, the target network. The target network is updated at each step from the Q-network by a weighted average leading to increased stability [29,30]. Another useful trick is to employ an ϵ-greedy policy in the training stage [28], which simply means that the action associated with the highest Q-network output value is not always chosen, but with ϵ likelihood, a random action is selected. This allows the neural network to explore a more extensive state space and to avoid becoming trapped in a locally optimal solution. Finally, as is common to many forms of machine learning, a variety of batch sampling methods, in this case a replay buffer of previously experienced transitions, (s, a, s'), can be used to assist in the efficiency and stability of the learning minimization stage [31]. For further details of the algorithmic structure, we refer the reader to Appendix A.

IV. COMPONENT DESIGN

We now use the Q-learning framework to design the essential elements for building an interferometer, i.e., a splitter,

reflectors, and a recombiner. One aspect that is worth emphasizing is that the standard Mach-Zehnder interferometer topology may not always be the optimal layout for determining an accumulated quantum phase. For example, it has been demonstrated in Ref. [32] that novel optical experimental setups that could potentially provide sensitivity surpassing that of the Mach-Zehnder configuration can be designed with reinforcement learning. However, with the aim of simplicity, and even though it constrains our solution space, we will limit the discussion here to the design of only the specified individual interferometer components. This will allow us to evaluate the efficacy of the learned protocols by direct comparison with the device characteristics of the analogous Bragg interferometer. In the future, it may prove beneficial to allow for more freedom in the design topology and to thereby ascertain whether this leads to further performance gains.

We have developed the learning terminology around words such as "state" and "environment" where they have been given a specific meaning. We are now going to apply the methodology to the design task of controlling a complex quantum system. Unfortunately we will encounter the issue that terms such as state and environment have conflicting and completely different meanings in the theory of open quantum systems. In particular, to enable an efficient machine learning algorithm with an addressable number of input nodes, the state space that we use is typically a compact representation of a quantum state or operator. The environment, which denotes the system that we wish to control and that is evolving under our imposed actions, is a completely different use of the term to that common in open quantum system literature. In the next design descriptions, we will provide the dictionary for mapping the learning terminology to the quantum variables.

A. Beam splitter

The beam splitter is a target-state design problem that can be specified as follows. We initialize a quantum state in the ground state of an optical lattice potential, where the lattice depth V_0 is set to $10\,E_r$ ($E_r = \hbar^2 k_L^2/2m$ is the recoil energy). The task is to shake the lattice in such a way as to split the atoms from the ground state into an approximate equal superposition of $|\pm p_0\rangle$ momentum states. We will focus on $p_0 = 4\hbar k_L$ here, which we find to balance well the trade-off between large momentum splitting and high-frequency components necessary for the shaking function. If the target quantum state is a lattice eigenstate it is a stationary state under free evolution, avoiding the need for a free evolution protocol. As a consequence, we assign the target quantum state to be the third-excited Bloch state with zero quasimomentum. This eigenstate has the attractive property that it closely approximates the desired superposition of momenta, although not perfectly, and can be strongly coupled to the ground state by a reasonably simple shaking solution.

The state space for learning, i.e., the vector space containing s, is a compact representation of the Hilbert space. For this learning task, we define s to be a vector of populations (i.e., probability distribution) for momenta chosen from a comb of discrete values, $\{2n\hbar k\}$ for $n \in \mathbb{Z}$, $|n| < n_{\max}$. This choice is motivated by the fact that the stimulated absorption and emission of photons only couple momenta separated by this quantized spacing. The possible actions, a, are also reduced to a discrete set of possibilities, each action corresponding to a specific constant phase to be applied during the corresponding step (see Appendix C). This means that with these associations, the deep Q-network is specified as mapping momentum populations at its input layer to Q-values at its output layer associated with a finite set of possible lattice phases to be applied during the next update.

In order to apply the action generated by a choice of lattice phase and to thereby determine the updated momentum distribution that results, we need to specify the environmental model. In principle, this could be experimental or theoretical, and even if theoretical, could include stochastic effects that arise from dissipation and noise. However, for simplicity, we will take the most basic formulation. This means that our environmental model will update a pure quantum state through the Schrödinger equation evolution for an isolated system. The reward will be based on the fidelity between the evolved quantum state and the target quantum state, i.e., the third-excited Bloch state, as calculated from the modulus-square inner product.

Actually, this environmental model allows us to accomplish two design tasks at the same time. Once we find a protocol for splitting, the recombining task can be achieved by simply implementing the time-reversed protocol, a consequence of the time-reversal symmetry underpinning the Schrödinger evolution and the time-reversal symmetry of both the initial and the target states.

A reinforcement learning outcome for the beam splitter is shown in Fig. 3, which illustrates the learned shaking function, $\phi(t)$, and the corresponding evolution for the momentum probability distribution that results. Even though this sequence of phases does not appear to have a predictable structure, it is apparent that at the terminal time the momentum distribution has the anticipated form of two well-defined peaks at $\pm 4\hbar k_L$. The calculated fidelity of the state is approximately 95%. It would be possible to optimize further by expanding the set of possible phase elements, by increasing the total evolution time, or by optimizing the learning cycles. However, any real experiment may possess imperfections and other aspects that are not well described by our isolated system's Schrödinger evolution, and the level of performance of our design is sufficient for the task at hand.

B. Mirror

The second task is to reflect the momentum from $|\pm p_0\rangle$ to $|\mp p_0\rangle$, that is, corresponding to a matter-wave mirror. We should point out at the start that this task is essentially different in character from the beam splitter, because it is not a target-state but a target-operator design problem. The mirror is defined by the desired map,

$$\alpha|p_0\rangle + \beta|-p_0\rangle \to \alpha|-p_0\rangle + \beta|p_0\rangle,$$

for any arbitrary α and β. This implies that the target unitary is any operator $\hat{U}_{\text{target}}$ that satisfies

$$\hat{P}\hat{U}_{\text{target}}\hat{P} = |p_0\rangle\langle -p_0| + |-p_0\rangle\langle p_0|, \tag{5}$$

where $\hat{P} = |p_0\rangle\langle p_0| + |-p_0\rangle\langle -p_0|$ is the projector onto the 2-dimensional subspace. The design goal is, therefore, to shake

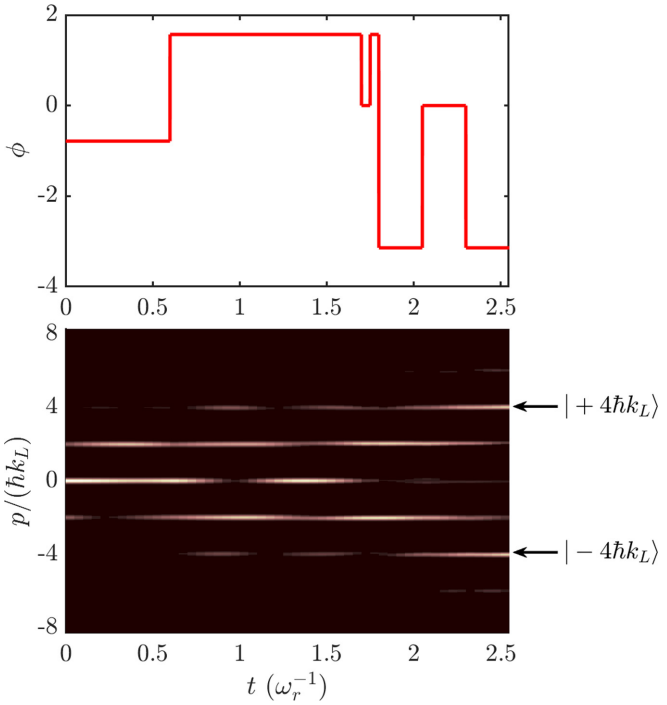

FIG. 3. Shaking function for splitting. The learned lattice phase, $\phi(t)$, (top) is allowed to take on one of a discrete set of five possible values that span the range shown. The momentum probability distribution (bottom) is initialized to the ground state of the lattice and at the terminal time well approximates the desired superposition for a beam splitter.

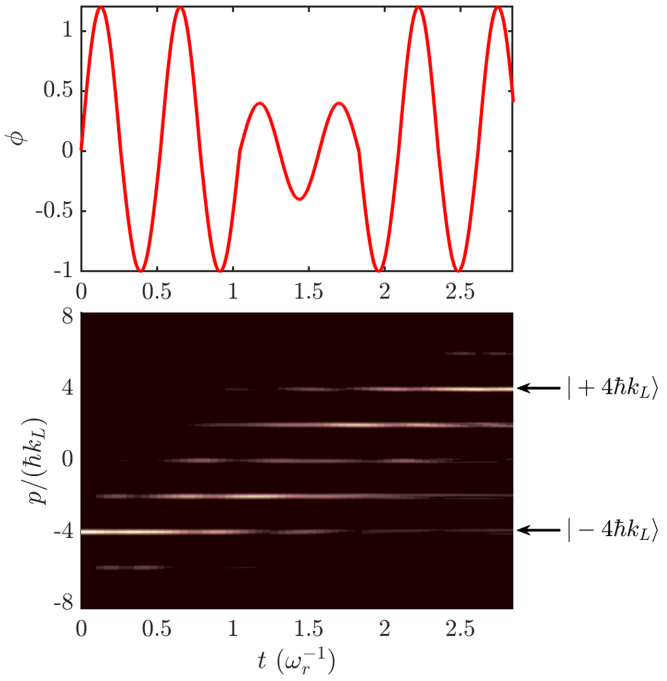

FIG. 4. Shaking function for reflection. The learned lattice phase, $\phi(t)$, (top) is sinusoidal and the amplitude is allowed to take on any one of a discrete set of five values at each half cycle. The momentum distribution (bottom) is prepared in $-4\hbar k_L$ and well approximates $4\hbar k_L$ at the terminal time.

the lattice as a function of time, such that the corresponding unitary evolution, $\hat{U}(t)$, approximates any $\hat{U}_{\text{target}}$ that satisfies Eq. (5) at the terminal time. The vector space containing s is defined as a compact representation, in this case, of the set of quantum unitary operators. Given the mirror behavior, a suitable choice for s for this learning task is a vector of norms of selected matrix elements of the unitary represented in momentum space, i.e., $|\langle \pm p_0|\hat{U}(t)|p = 2n\hbar k\rangle|^2$ for $n \in \mathbb{Z}$, $|n| < n_{\max}$. For the set of possible actions, a, we use a different parametrization of the phase than for the beam splitter design. We found that the quality of the mirror is very sensitive to the frequency of the shaking function. The characteristic frequency, $12\omega_r$, where $\omega_r = \hbar k_L^2/2m$ is known as the recoil frequency of the lattice, corresponds to the kinetic energy difference between $|4\hbar k_L\rangle$ and $|2\hbar k_L\rangle$ and works especially well. Consequently, we enforce a sinusoidal $\phi(t)$ at frequency $12\omega_r$, and at each half cycle, the agent chooses the action a as the amplitude of the sinusoidal shaking function selected from a small set of possible values (see Appendix D). This means here that the deep Q-network is specified as mapping a compact representation formed from the norm of a subset of elements of the unitary matrix to Q-values associated with the amplitude of the sinusoidal shaking function for its next half cycle.

We also need to define what is meant by the environment in this case. The environmental model establishes the unitary operator update during a shaking function half cycle. If, as before, we take the environmental model to be a closed quantum system, the evolution is given by solving the Schrödinger equation of motion for the time evolution operator,

$$i\hbar\frac{d\hat{U}(t)}{dt} = \hat{H}(t)\hat{U}(t). \tag{6}$$

The reward is associated with the channel fidelity in the $d_s =$ 2-dimensional subspace spanned by $|\pm p_0\rangle$ (see [33])

$$\mathcal{F} = \frac{1}{d_s(d_s+1)}[\text{Tr}(MM^\dagger) + |\text{Tr}(M)|^2], \tag{7}$$

where the density operator is $M = PU_{\text{target}}^\dagger U(T)P$.

An example outcome from reinforcement learning for the mirror is shown in Fig. 4. This illustrates the learned shaking function and corresponding momentum probability density that this generates starting from $-4\hbar k_L$. From matrix elements of the resulting unitary operator, we also verify that the phase relation between the coefficients of the $\pm 4\hbar k_L$ states is preserved. As for the beam splitter case, it would be difficult to anticipate the form of $\phi(t)$, and yet the resulting quantum state evolution closely approximates that expected for a mirror. The channel fidelity reaches 93% at the terminal time, which as discussed earlier could still be improved on, but nevertheless is sufficient for our purpose.

C. Matter-wave interferometer

We can now cascade together the learned components to make a shaken lattice interferometer. In between the components, we allow the system to propagate freely with the lattice present but with $\phi(t) = 0$. This gives a time evolution of the wave function in real space as shown in Fig. 5, where the state has been initialized to be in the ground state of the

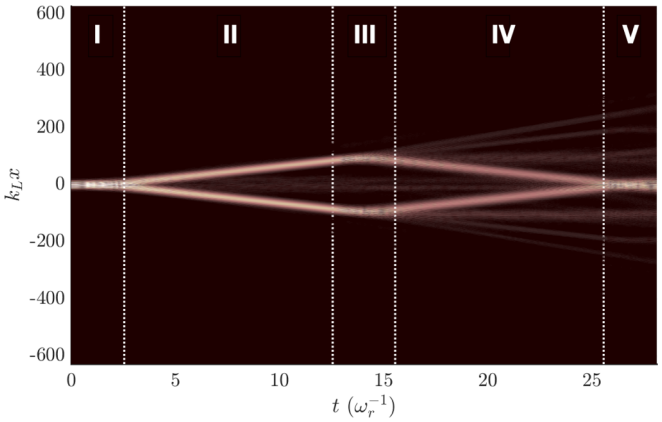

FIG. 5. Time evolution of the matter-wave density throughout the entire interferometry sequence. The white dotted lines separate the plot into five regions. In region I, we apply the splitting protocol. In region II, we allow the matter wave to propagate freely in the lattice. The appearance of two wave packets traveling in opposite directions shows that the beam splitter operates as expected. In region III, we apply the mirror shaking function. The matter wave undergoes free propagation in region IV again, and the two wave packets switch directions, demonstrating the functionality of the mirror. Last, we apply the recombining protocol in region V. Apart from the main closed diamond-shaped paths, we observe that there are auxiliary paths that are fainter but still clearly evident. They arise due to the imperfection of the components and also due to the side peaks arising from the third-excited Bloch state.

lattice multiplied in real space by a Gaussian envelope with a width of a few lattice sites. The free-propagation time is set to $10\,\omega_r^{-1}$. During free propagation, we can see the wave packets propagating at the anticipated $\pm 4\hbar k_L/m$ group velocity. The anticipated features are evident in the observed splitting, reflection, and recombination of the matter wave.

V. STATISTICAL ANALYSIS

We compute the final momentum distribution on a grid of acceleration values in order to see how useful our device is for inertial sensing. To interpret the outcome of the interferometer, we use Bayes theorem to derive information about the acceleration from the momentum distribution. This approach is always more effective than curve-fitting when there is knowledge of the probability distribution generator, and the difference is particularly notable in situations like this when the distributions are multimodal. The probability distribution of the acceleration after measurement of a particle in a particular momentum state, $P(a|p)$, is given by,

$$P(a|p) \propto P(p|a)P(a), \tag{8}$$

where $P(a)$ is the prior distribution of the acceleration, and the proportionality is resolved by normalizing the total probability to unity. The probability for measuring a particular momentum p conditioned on the acceleration a, $P(p|a)$ is directly calculated by propagating the wave function in time. We combine multiple measurements by iterating Eq. (8) to formulate the conditional probability distribution based on the entire measurement record [34]. For a measurement record $\{p_1, p_2, \ldots, p_N\}$, where N is the number of measurements,

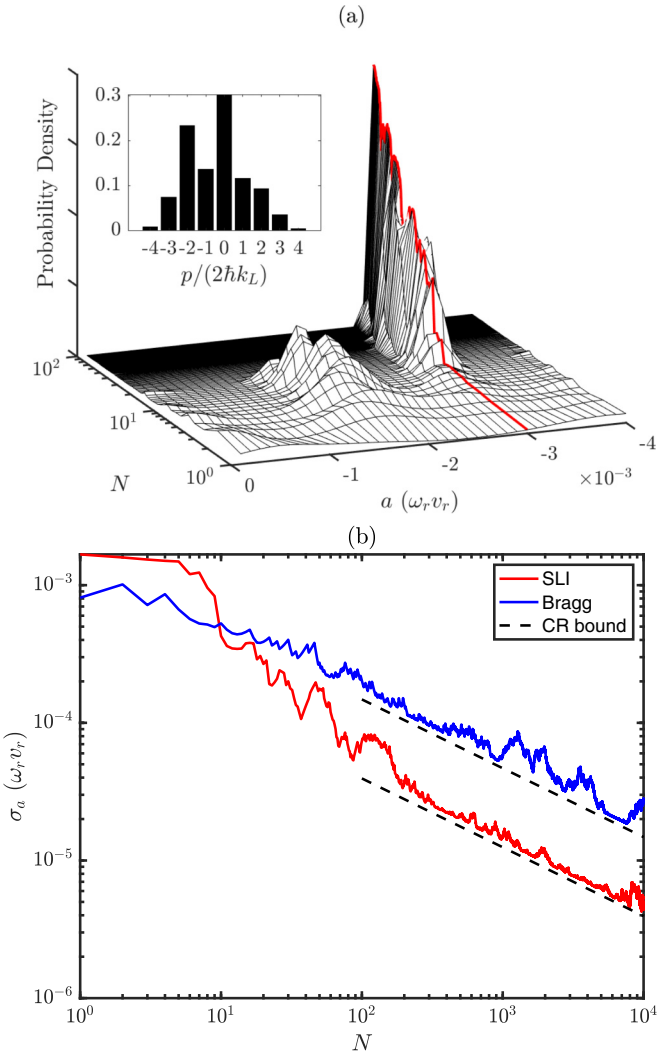

FIG. 6. (a) Posterior probability density of the acceleration for the first 100 atoms. The true acceleration that we aim to reveal by the measurements is $-3 \times 10^{-3}\,\omega_r v_r$ (red line), and the measurements are sampled from the momentum distribution at the end of the interferometry sequence, as shown in the inset. (b) Standard deviation of the acceleration estimated using Bayes theorem for up to 10^4 atoms. We show the results from both the shaken lattice interferometer (SLI) and the Bragg interferometer and conclude that the SLI has a higher sensitivity. The standard deviations σ_a are roughly inversely proportional to the square root of the number of measurements N, for $N > 10^2$. The black dashed lines are the Cramér-Rao (CR) lower bounds and scale exactly as $1/\sqrt{N}$.

the probability distribution becomes

$$P(a|p_1, \ldots, p_N) \propto P(p_N|a) \ldots P(p_1|a)P(a), \tag{9}$$

normalized to unity. Since we assume that each atom is independent of each other, N is the total number of atoms we observe. We combine the distributions for each atom using Eq. (9) to accumulate the distribution for a conditioned on the measurement record and estimate the acceleration by taking the expectation value.

In Fig. 6(a), we show an example of the parameter estimation process, where the actual acceleration is $-3 \times 10^{-4}\,\omega_r v_r$ ($v_r = \hbar k_L/m$ is the recoil velocity). The corresponding mo-

mentum probability distribution for each atom is shown in the inset. What is important about this distribution is that it is almost unique for each acceleration value and therefore acts as a fingerprint. In Figure 6(b), we show the standard deviation of the acceleration as a function of the number of measurements. A typical number of atoms for an ultracold gas experiment may be of order $\sim 10^4$, and with 10^4 measurements, the estimated acceleration agrees extremely well with the actual value. We observe that the standard deviation approaches the standard quantum limit, which scales as $\sim 1/\sqrt{N}$. This result is consistent with the fact that we are not considering atom interactions in the system, so each atom may be thought of as making an independent measurement.

We demonstrate that reinforcement learning provides additional capability over traditional experimental techniques by comparing the sensitivity of the shaken lattice interferometer with the conventional Bragg interferometer. The Bragg interferometry sequence is shown in Fig. 1(b). We use the Bayesian parameter estimation result with the Bragg interferometer as a baseline for benchmarking the shaken lattice interferometer. To make a fair comparison, we consider the case where the Bragg interferometry sequence takes the same time as the shaken lattice interferometer. The protocol found by our reinforcement learning algorithm generates a momentum splitting of $8\hbar k_L$ (between $-4\hbar k_L$ and $+4\hbar k_L$) at the beam splitter, four times larger than the $2\hbar k_L$ (between $-1\hbar k_L$ and $+1\hbar k_L$) splitting from Bragg diffraction, and therefore this results in higher sensitivity. One may argue that it is possible to generate larger momentum splitting with Bragg diffraction, and while this is true, higher-order transitions typically require much more time than than the short pulses applied here. For these reasons, we have observed from our simulations that the standard deviation of the shaken lattice interferometer can be approximately four times lower than that of the corresponding Bragg interferometer given the time constraint, implying that we have realized an approximate factor of four in sensitivity gain. The results are shown in Fig. 6(b).

We verify that the standard deviations σ_a for both the shaken lattice interferometer and the Bragg interferometer are close to the limits set by the Cramér-Rao bound, which can be calculated from the classical Fisher information as,

$$\sigma_a \geqslant 1/\sqrt{N I_1(a)},$$
$$I_1(a) = \sum_p \frac{1}{P(p|a)} \left[\frac{\partial P(p|a)}{\partial a}\right]^2, \tag{10}$$

where $I_1(a)$ is the Fisher information for one independent measurement. The Cramér-Rao bound for the shaken lattice interferometer is determined by numerically calculating the classical Fisher information using the precalculated probability $P(p|a)$, and the Cramér-Rao bound for the Bragg interferometer is determined by the analytical form $\sigma_a \geqslant (2k_L T^2)^{-1} N^{-1/2}$, where T is the free-propagation time.

Note that the sensitivity presented here is not the ultimate achievable sensitivity since we are constraining system parameters. It would be possible to improve the sensitivity of our interferometer, for example, by increasing the free-propagation time, or by further accelerating the matter-wave components in opposite directions after applying our splitting protocol, as discussed in Refs. [36,37].

VI. CONCLUSION

In summary, we have demonstrated a machine learning methodology to control a complex quantum system for the purpose of performing a quantum metrology task. Specifically, we demonstrated how to utilize reinforcement learning for the design of a lattice-based matter-wave interferometer. We showed that by shaking the lattice with protocols derived from deep Q-learning, matter-wave analogs of optical components are realized. We showed that these can be concatenated together to build a high-precision interferometric device. The multipath interferometer that was constructed in this way is capable of the measurement of acceleration with higher sensitivity than that achieved by a conventional matter-wave interferometer.

While we have assumed that atom interactions are negligible, they could potentially bring more quantum advantage if harnessed appropriately. For example, if we could discover a protocol for shaking the lattice to generate momentum-squeezed states, the sensitivity could potentially be higher than the standard quantum limit that is generated by the statistical averaging of independent atoms. In fact, the quantum entanglement in such squeezed states could allow the resulting interferometer to approach the ultimate limit to sensitivity set by the Heisenberg uncertainty relation between number and phase. We have focused on demonstrating a specific solution to the problem of designing a Mach-Zehnder interferometer, but our main outcome is actually to illustrate an effective design methodology. In the future, this approach could potentially lead to the design of alternate protocols that construct completely new types of metrological devices with a performance level surpassing conventional experimental methods.

ACKNOWLEDGMENTS

The authors acknowledge helpful discussions with Dana Anderson, Marco Nicotra, Claire Monteleoni, Haonan Liu, and Simon Jäger. This work was supported by NSF Grant No. 1936303, NSF QLCI Award No. OMA-2016244, NSF PFC Grant No. 1734006, and the DARPA and ARO Grant No. W911NF-16-1-0576.

APPENDIX A: TRAINING THE DEEP Q-NETWORK

The objective of deep Q-learning is to train the deep Q-network so that it represents the return defined in Eq. (2) well and therefore satisfies the Bellman optimality equation,

$$Q(s,a) = r(s') + \gamma \max_{a'} Q(s',a'). \tag{A1}$$

We train the deep Q-network using the double deep Q-learning algorithm [30]. In this algorithm, an extra network called the target network is included to estimate the future return [Eq. (3)]. The reason to do it this way is that Eq. (A1) involves a self-consistency element. If during the training stage the Q-values on both sides of the equation are determined by the same Q-network, then updating the Q-network will simultaneously change both the Q-value $Q(s,a)$ and the

Algorithm 1: Double Deep Q-Learning

Initialize Q-network $Q_{\boldsymbol{\theta}}$ and target network $Q_{\boldsymbol{\theta}'} \leftarrow Q_{\boldsymbol{\theta}}$;
for *trajectory=1:episodes* **do**
  $s \leftarrow s_0$;
  while *d=0* **do**
    if $rand() > \epsilon$ **then**
      $a \leftarrow \arg\max_a Q_{\boldsymbol{\theta}}(s, a)$;
    else
      $a \leftarrow rand(\mathcal{A})$;
    end
    $[s', r, d] \leftarrow environment(s, a)$;
    Store (s, a, s', r, d) in replay buffer;
    Sample from replay buffer:
     $\{(s_i, a_i, s_i', r_i, d_i) | \, i = 1, ..., B\}$;
    $\mathbb{L} = \frac{1}{B}\sum_i [Q_{\boldsymbol{\theta}}(s_i, a_i) - Y_i]^2$,
     $Y_i = r_i + \gamma Q_{\boldsymbol{\theta}'}(s_i', \arg\max_{a'} Q_{\boldsymbol{\theta}}(s_i', a_i'))$;
    Update $Q_{\boldsymbol{\theta}}$: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathbb{L}$;
    Update $Q_{\boldsymbol{\theta}'}$: $\boldsymbol{\theta}' \leftarrow \tau\boldsymbol{\theta}' + (1-\tau)\boldsymbol{\theta}$;
    $s \leftarrow s'$;
  end
end

return $Y(s')$ that the Q-value is supposed to converge to. This creates a feedback cycle that can make the learning process extremely unstable, and in fact the Q-values may never converge. In order to mitigate this problem, we employ a Q-network, $Q_{\boldsymbol{\theta}}$, just as described, from which the optimal action is chosen by the agent through their policy. However, we then update the return used to optimize the Q-network with a Q-value corresponding to this action but obtained from a second neural network, the target network, $Q_{\boldsymbol{\theta}'}$. The target network is only updated more slowly through a weighted average. This avoids the unstable feedback since the Q-network and target network are weakly correlated at early stages of the learning process [29]. Furthermore, this method also resolves issues with overoptimism in the standard deep Q-learning algorithm [30]. A sketch of the algorithm is presented in Algorithm 1.

The algorithm is described as follows. First, we initialize the weights and biases in the Q-network randomly and initialize the target network such that its network parameters are the same as those of the Q-network. Then, we iterate over a large number of episodes. In each episode, a trajectory of consecutive states and actions is generated. A trajectory starts with an initial state s_0 and ends when we reach a terminal state or a threshold number of steps.

In each step, we choose the action that maximizes the Q-value for the current state most of the time, but with probability ϵ we choose a random action. In practice, ϵ decays from 1 to 0.01 over time, since exploitation becomes more important than exploration as the learning process proceeds. The action is then fed into the environment, and the environment outputs the subsequent state s', a reward r, and a Boolean d that indicates whether the trajectory terminates at this step. A tuple (s, a, s', r, d), which represents the experience we get from taking this step, is pushed into the replay buffer.

To minimize the error of $Q_{\boldsymbol{\theta}}(s, a)$ for representing the return, we calculate a loss function defined as the squared difference between $Q_{\boldsymbol{\theta}}(s, a)$ and $Y(s')$. The future return in $Y(s')$ [Eq. (3)] is evaluated by the output value of the target network for the next optimal action. The next optimal action is determined by the action that maximizes the output values of the Q-network, given the next state s' as input. Since we need the Q-values to do well for all possible pairs of (s, a), we draw B (batch size) samples from the replay buffer, where each sample is labeled by an index i, and calculate their average loss.

The parameters in the Q-network are updated through gradient descent in the direction of decreasing average loss. The parameters in the target network are updated by taking the weighted average of the previous parameters from the target network and the updated parameters from the Q-network.

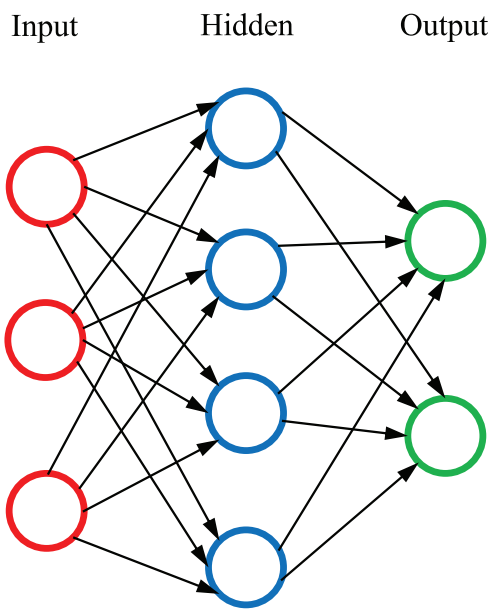

FIG. 7. An example topology of the network that we used in the deep Q-algorithm.

APPENDIX B: NETWORK TOPOLOGY

The Q-network we use includes an input layer, one hidden layer, and an output layer. Each layer is connected to the next one by a linear function followed by a ReLU (Rectified Linear Unit) activation function, which represents the function $\max(x, 0)$ with the argument x being a real number. The relations between layers are explicitly written out as follows:

$$\boldsymbol{h} = \mathrm{ReLU}(\mathbf{W}_1\boldsymbol{x} + \boldsymbol{b}_1) \tag{B1}$$

$$\boldsymbol{y} = \mathrm{ReLU}(\mathbf{W}_2\boldsymbol{h} + \boldsymbol{b}_2), \tag{B2}$$

where $\boldsymbol{h}$ denotes the hidden nodes, $\boldsymbol{x}$ denotes the input nodes, $\boldsymbol{y}$ denotes the output nodes, $\mathbf{W}_1$, $\mathbf{W}_2$ are the matrices of weights, and $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ are vectors of biases. An example topology of the Q-network is shown in Fig. 7.

APPENDIX C: REINFORCEMENT LEARNING FOR SPLITTING

Here we present how we formulate the control for splitting as a reinforcement learning problem. First of all, we define the input state to be a vector of the populations in the momentum

TABLE I. Choice of hyperparameters for the splitting task. The exploration probability ϵ decays linearly for each episode at the rate represented by ϵ decay. The optimizer we use, "Adam" (which stands for adaptive momentum estimation), is a method for efficient stochastic gradient descent [35].

Hyperparameters	Values
γ	0.999
τ	0.999
α	0.001
episodes	20,000
ϵ decay	0.0001
Hidden size	98
Batch size	64
Optimizer	Adam

eigenstates,

$$\{|2n\hbar k_L\rangle \mid n = -3, -2, -1, 0, 1, 2, 3\}. \tag{C1}$$

The momentum states that are neglected here are barely populated the whole time. The actions are chosen from a set of discrete phase values,

$$\{n\pi/4 \mid n = -4, -2, -1, 0, 2\}. \tag{C2}$$

The reward is a function of fidelity, defined as

$$r = \begin{cases} 0 & \text{if } d = 0 \\ \frac{\mathcal{F}}{1-\mathcal{F}} & \text{if } d = 1 \end{cases}, \tag{C3}$$

$$\mathcal{F} = |\langle\psi_{\text{target}}|\psi(t)\rangle|^2, \tag{C4}$$

where $d = 1$ when the maximum terminal time is reached or when the fidelity is higher than the threshold 0.95, and $d = 0$ otherwise. The Q-network we use for this task has an input dimension of 7 and an output dimension of 6, as a result of (C1) and (C2). The neural network is implemented in PyTorch. The values for the hyperparameters used for training this Q-network are listed in Table I.

APPENDIX D: REINFORCEMENT LEARNING FOR THE MIRROR

The reflection operation is defined by a target unitary operator that satisfies Eq. (5). Our goal is to control the phase as a function of time, such that the unitary operator,

$$\hat{U}(t) = \mathcal{T}\left\{\exp\left[\int_0^t -i\hat{H}(\phi(t'))dt'/\hbar\right]\right\}, \tag{D1}$$

reaches the target operator at the terminal time. Here $\mathcal{T}$ represents the time-ordering operator and $\hat{H}(\phi)$ is defined in Eq. (1). To formulate the control problem as a reinforcement learning problem, we assign the state to be the matrix elements of the unitary operator. Since before the wave function enters the reflection stage, most of its population is in the $|\pm 4\hbar k_L\rangle$ subspace, we only keep track of the operation in this subspace. Because of this, we reduce the input state to the modulus square of the matrix elements for only

$$\{|2n\hbar k_L\rangle\langle\pm 4\hbar k_L||n = -3, -2, -1, 0, 1, 2, 3\}. \tag{D2}$$

The state is chosen so that it forms an experimental observable. This state can be obtained by initializing the quantum state to one of the $|\pm 4\hbar k_L\rangle$ states, applying the control, and measuring the populations for all the momentum states through time-of-flight imaging.

As discussed in the main text, we found that the frequency of the shaking function is important for the performance of the mirror. Simply by modulating at the characteristic frequency, $12\omega_r$, which corresponds to the energy difference between $|4\hbar k_L\rangle$ and $|2\hbar k_L\rangle$, with a fixed amplitude for a duration of $\sim 3\omega_r^{-1}$, the channel fidelity can reach as high as 0.8. We take advantage of this knowledge, and try to improve the fidelity on the fixed-amplitude modulation. We choose our actions to be the amplitude of the sinusoidal modulation to the phase, so the resulting phase is $\phi(t) = \text{Amp}(t) \times \sin(12\omega_r t)$. The amplitudes are chosen from a discrete set of values,

$$\text{Amp} = \{0.4,\ 0.6,\ 0.8,\ 1.0,\ 1.2\}. \tag{D3}$$

The time interval between each decision point is $\pi/12 \approx 0.26\,\omega_r^{-1}$, and in each interval the amplitude is held constant.

The reward function is the same as Eq. (C3), except that the fidelity $\mathcal{F}$ is now defined with the channel fidelity in the relevant subspace [see Eq. (7)].

With the states, actions, rewards specifically designed for reflection, we use the double deep Q-learning algorithm to learn the strategy for controlling the amplitude of the sinusoidal modulation to the phase. The deep Q-network used for learning the mirror has an input size of 14 and an output size of 5, as a result of (D2) and (D3). The hyperparameters are listed in Table II.

TABLE II. Choice of hyperparameters for the reflection task.

Hyperparameters	Values
γ	0.99
τ	0.99
α	0.001
ϵ decay	0.0005
episodes	8,000
Hidden size	128
Batch size	32
Optimizer	Adam

[1] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), Observation of Gravitational Waves from A Binary Black Hole Merger, Phys. Rev. Lett. **116**, 061102 (2016).

[2] R. H. Parker, C. Yu, W. Zhong, B. Estey, and H. Müller, Measurement of the fine-structure constant as a test of the standard model, Science **360**, 191 (2018).

[3] S. Fray, C. A. Diez, T. W. Hänsch, and M. Weitz, Atomic Interferometer with Amplitude Gratings of Light and its

Applications to Atom Based Tests of the Equivalence Principle, Phys. Rev. Lett. **93**, 240404 (2004).

[4] T. L. Gustavson, P. Bouyer, and M. A. Kasevich, Precision Rotation Measurements with an Atom Interferometer Gyroscope, Phys. Rev. Lett. **78**, 2046 (1997).

[5] H. F. Rice, V. Benischek, and L. Sczaniecki, Application of atom interferometric technology for gps independent navigation and time solutions, in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA* (IEEE, PIscataway, NJ, 2018), pp. 1097–1106.

[6] A. P. Peirce, M. A. Dahleh, and H. Rabitz, Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications, Phys. Rev. A **37**, 4950 (1988).

[7] J. A. Nelder and R. Mead, A simplex method for function minimization, Comput. J. **7**, 308 (1965).

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).

[9] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nature (London) **529**, 484 (2016).

[10] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, Mastering the game of go without human knowledge, Nature (London) **550**, 354 (2017).

[11] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, A general reinforcement learning algorithm that masters chess, shogi, and go through self-play, Science **362**, 1140 (2018).

[12] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement Learning in Different Phases of Quantum Control, Phys. Rev. X **8**, 031086 (2018).

[13] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Reinforcement Learning with Neural Networks for Quantum Feedback, Phys. Rev. X **8**, 031084 (2018).

[14] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, Optimizing quantum error correction codes with reinforcement learning, Quantum **3**, 215 (2019).

[15] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, Universal quantum control through deep reinforcement learning, npj Quantum Inf. **5**, 33 (2019).

[16] M. Bukov, Reinforcement learning for autonomous preparation of floquet-engineered states: Inverting the quantum kapitza oscillator, Phys. Rev. B **98**, 224305 (2018).

[17] J. Mackeprang, D. B. R. Dasari, and J. Wrachtrup, A reinforcement learning approach for quantum state engineering, Quantum Mach. Intell. **2**, 5 (2020).

[18] A. A. Melnikov, P. Sekatski, and N. Sangouard, Setting up Experimental Bell Tests with Reinforcement Learning, Phys. Rev. Lett. **125**, 160401 (2020).

[19] J. Wallnöfer, A. A. Melnikov, W. Dür, and H. J. Briegel, Machine learning for long-distance quantum communication, PRX Quantum **1**, 010301 (2020).

[20] J. Schuff, L. J. Fiderer, and D. Braun, Improving the dynamics of quantum sensors with reinforcement learning, New J. Phys. **22**, 035001 (2020).

[21] L. Mach, Ueber einen interferenzrefraktor, Z. Instrumentenkd. **12**, 89 (1892).

[22] L. Zehnder, Ein neuer interferenzrefraktor, Z. Instrumentenkd. **11**, 275 (1891).

[23] C. Bordé, Atomic interferometry with internal state labelling, Phys. Lett. A **140**, 10 (1989).

[24] M. Kasevich and S. Chu, Atomic Interferometry Using Stimulated Raman Transitions, Phys. Rev. Lett. **67**, 181 (1991).

[25] C. A. Weidner, H. Yu, R. Kosloff, and D. Z. Anderson, Atom interferometry using a shaken optical lattice, Phys. Rev. A **95**, 043624 (2017).

[26] C. A. Weidner and D. Z. Anderson, Experimental Demonstration of Shaken-Lattice Interferometry, Phys. Rev. Lett. **120**, 263201 (2018).

[27] C. A. Weidner and D. Z. Anderson, Simplified landscapes for optimization of shaken lattice interferometry, New J. Phys. **20**, 075007 (2018).

[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT press, Boston, MA, 2018).

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, Human-level control through deep reinforcement learning, Nature (London) **518**, 529 (2015).

[30] H. Van Hasselt, A. Guez, and D. Silver, Deep reinforcement learning with double q-learning, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (AAAI Press, Phoenix, Arizona, 2016), pp. 2094–2100.

[31] L.-J. Lin, Self-improving reactive agents based on reinforcement learning, planning and teaching, Mach. Learn. **8**, 293 (1992).

[32] A. A. Melnikov, H. Poulsen Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Active learning machine learns to create new quantum experiments, Proc. Natl. Acad. Sci. U.S.A. **115**, 1221 (2018).

[33] L. H. Pedersen, N. M. Møller, and K. Mølmer, Fidelity of quantum operations, Phys. Lett. A **367**, 47 (2007).

[34] M. J. Holland and K. Burnett, Interferometric Detection of Optical Phase Shifts at the Heisenberg Limit, Phys. Rev. Lett. **71**, 1355 (1993).

[35] D. Kingma and J. Ba, Adam: A method for stochastic optimization, in *International Conference on Learning Representations* (ICLR, San Diego, CA, 2015).

[36] M. Gebbe, J.-N. Siemß, M. Gersemann, H. Müntinga, S. Herrmann, C. Lämmerzahl, H. Ahlers, N. Gaaloul, C. Schubert, K. Hammerer, S. Abend, and E. M. Rasel, Twin-lattice atom interferometry, Nat. Commun. **12**, 2544 (2021).

[37] Z. Pagel, W. Zhong, R. H. Parker, C. T. Olund, N. Y. Yao, and H. Müller, Symmetric bloch oscillations of matter waves, Phys. Rev. A **102**, 053312 (2020).