Adaptive scheduling of noise characterization in quantum computers

Riddhi Swaroop Gupta,* Alistair R. Milne, Claire L. Edmunds, Cornelius Hempel, and Michael J. Biercuk
ARC Centre of Excellence for Engineered Quantum Systems, School of Physics, The University of Sydney, New South Wales 2006, Australia

New quantum computing architectures consider integrating qubits as sensors to provide actionable information useful for decoherence mitigation on neighboring data qubits. Little work has addressed how such schemes may be efficiently implemented in order to maximize information utilization when noise fields possess long-range correlations. We present an autonomous learning framework, Quantum Simultaneous Localization and Mapping (QSLAM), for adaptive scheduling of sensor-qubit measurements and efficient (in measurement number and time) spatial noise mapping across device architectures. Via a two-layer particle filter, QSLAM receives binary measurements and determines regions within the architecture that share common noise processes; an adaptive controller then schedules future measurements to reduce map uncertainty. Numerical analysis and experiments on an array of trapped ytterbium ions demonstrate that QSLAM outperforms brute-force mapping by up-to 18x (3x) in simulations (experiments), calculated as a reduction in the number of measurements required to map a spatially inhomogeneous magnetic field with a target fidelity. As an early adaptation of robotic control to quantum devices, this work opens up exciting new avenues in quantum computer science.

I. INTRODUCTION

In the NISQ era [1], the impacts of decoherence and hardware error remain dominant considerations in achieving useful performance in quantum computations with realistic multi-qubit architectures [2–7]. In addition to techniques designed to suppress and correct errors through quantum control, there is substantial interest in deploying qubits as sensors at the physical level to provide actionable information on decoherence mechanisms, such as real-time measurements of ambient field fluctuations [8]. The objective is to exploit spatial correlations in long-range fields to detect noise on a so-called spectator qubit while actuating to stabilize proximal qubits [9] used in the computation [10]. Such spectator qubits may be either dedicated to this task or dynamically allocated as they become idle in a specific computation. Overall, this approach is a "spatially multiplexed" analogue of alternative control techniques which separate the tasks of detection and actuation in time [11, 12].

Even in medium-scale architectures subject to classical noise fields [13], inhomogeneities and temporal variations present a new requirement to spatially map the underlying noise processes in order to determine which qubits may be actuated upon using information from a specific sensor. Brute-force approaches are known to be inefficient, and it is desirable to seek automated mechanisms for physical noise mapping in quantum computing architectures that complement other autonomous optimization routines in order to improve overall hardware performance [14–18]. Fortunately, the conceptually related problem of mapping an unknown landscape is well studied in classical autonomous navigation and robotic exploration through a suite of algorithms known as Simultaneous Localization and Mapping (SLAM) [19–27].

* riddhi.sw@gmail.com

In this manuscript, we introduce a new framework for autonomous learning, denoted Quantum Simultaneous Localization and Mapping (QSLAM), to efficiently reconstruct a map of unknown spatial fields in a quantum computer. In our implementation, QSLAM discovers spatial 'neighborhoods' around spectator qubits in which information may be shared, representing the presence of spatial correlations in the underlying noise source. We implement the QSLAM framework via a novel two-layer particle filter and an autonomous real-time controller. Our algorithm iteratively builds a map of the noise field in real-time by maximizing the information utility obtained from each physical measurement. This, in turn, enables the controller to adaptively determine the highest-value measurement to be performed in the following step. We evaluate the performance of this algorithm in realistic test cases by both numeric simulation on 1D and 2D qubit arrays, and application to real experimental data derived from Ramsey measurements on a 1D crystal of trapped ions. Our results demonstrate that QSLAM outperforms brute-force measurement strategies by a reduction of up to 18× (3×) in the number of measurements required to estimate a noise map with a target fidelity for simulations (experiments). These results hold for both 1D and 2D qubit arrays subject to noise fields with varying spatial configurations.

II. THE QSLAM FRAMEWORK

We consider a spatial arrangement of d qubits as determined by a particular choice of hardware. An unknown, classical field exhibiting spatial correlations extends over all qubits on our notional device, and corresponds to ambient noise fields in realistic operating architectures. Our objective is to build a map of the underlying spatial variation of the noise field with the fewest possible single-qubit measurements, performed sequentially. We conceive that

each measurement is a single-shot Ramsey-like experiment in which the presence of the unknown field results in a measurable phase shift between a qubit's basis states at the end of a fixed interrogation period. This phase is not observed directly, but rather inferred from data, as it parameterizes the Born probability of observing a "0" or "1" outcome in a projective measurement on the qubit; our algorithms take these discretized binary measurement results as input data. The desired output at any given iteration t is a map of the noise field, denoted as a set of unknown qubit phases, F_t, inferred from the binary measurement record up to t.

The simplest approach to the mapping problem is to undertake a brute-force, "naive" strategy in which one uniformly measures sensor-qubits across the array, depicted schematically in Fig. 1(a). By extensively sampling qubit locations in space repeatedly, one can build a map of the underlying noise fields through the collection of large amounts of measurement information. Evidently, the naive, brute-force measurement approach is highly inefficient as it fails to recognize and exploit spatial correlations in the underlying noise field that may exist over a length-scale that exceeds the inter-qubit spacing. This is a particular concern in circumstances where qubit measurements are a costly resource *e.g.* in time, classical communications bandwidth, or qubit utilization when sensors are dynamically allocated.

The QSLAM framework shown in Fig. 1(b) stands in stark contrast. Here the underlying spatial correlations of the noise are mapped using a learning procedure, which adaptively determines the location of the next, most-relevant measurement to be performed based on past observations. With this approach, QSLAM reduces the overall resources required for spatial noise mapping by actively exploiting the spatial correlations to decide whether measurements on specific qubits provide an information gain.

We approximate the theoretical non-linear filtering problem embodied by QSLAM using particle-filtering techniques. Particle filters are used to propagate and update classical probability distributions in cases where they are expected to undergo highly non-linear transformations (cf. standard particle filters [28] and their use in classical SLAM implementations [29–31]). In our application, the choice of a particle filter to implement core QSLAM functionality accommodates the non-linear measurement model associated with the discretized outcomes given by projective qubit readout.

In our particle filtering implementation, at every iteration t, the information about the true map is contained in the state vector, X_t. Here the word 'state' refers to all quantities being inferred statistically from data, as opposed to an actual physical quantum state. Aside from the true map, the state vector, X_t, additionally contains information which approximates spatial gradients on the true map. The probability distribution of X_t conditioned on data is called the posterior distribution and it summarizes our best knowledge of the true map and its approximate spatial gradient information given past measurement outcomes. The posterior distribution is approximately represented as a discrete collection of weighted 'particles'. Each particle has two properties: a position and a weight. The position of the particle carries information about X_t, and the weight specifies the likelihood or the importance of the particle in the estimation procedure. All weights are initially equal at step $t = 0$ and after receiving measurement data at each step, the particles are 're-sampled'. This means that the original set

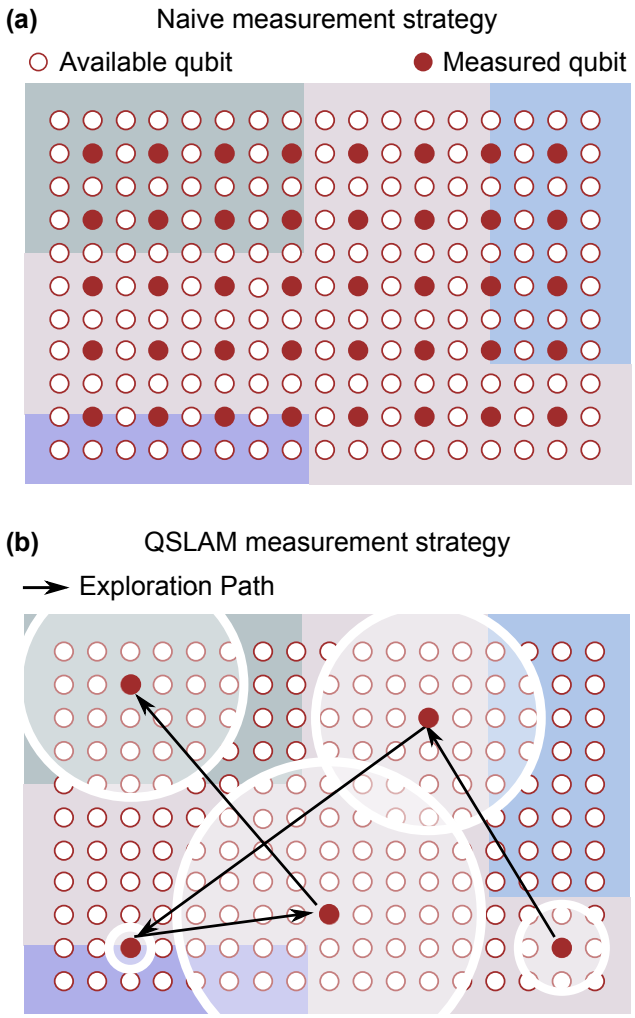

FIG. 1. Difference between a naive brute force and the QSLAM measurement strategy in reconstructing an inhomogenous background field. An arbitrary spatial arrangement of qubits (red circles) is shown with a true unknown field with finite correlations (colored regions). (a) The naive strategy measures the field across the array using a regular grid (red filled circles). (b) The QSLAM strategy iteratively chooses which qubit to measure next (black arrows) and additionally stores state estimation information in a form shared across local neighborhoods (white shaded circles), which reflects the spatial characteristics of the underlying map.

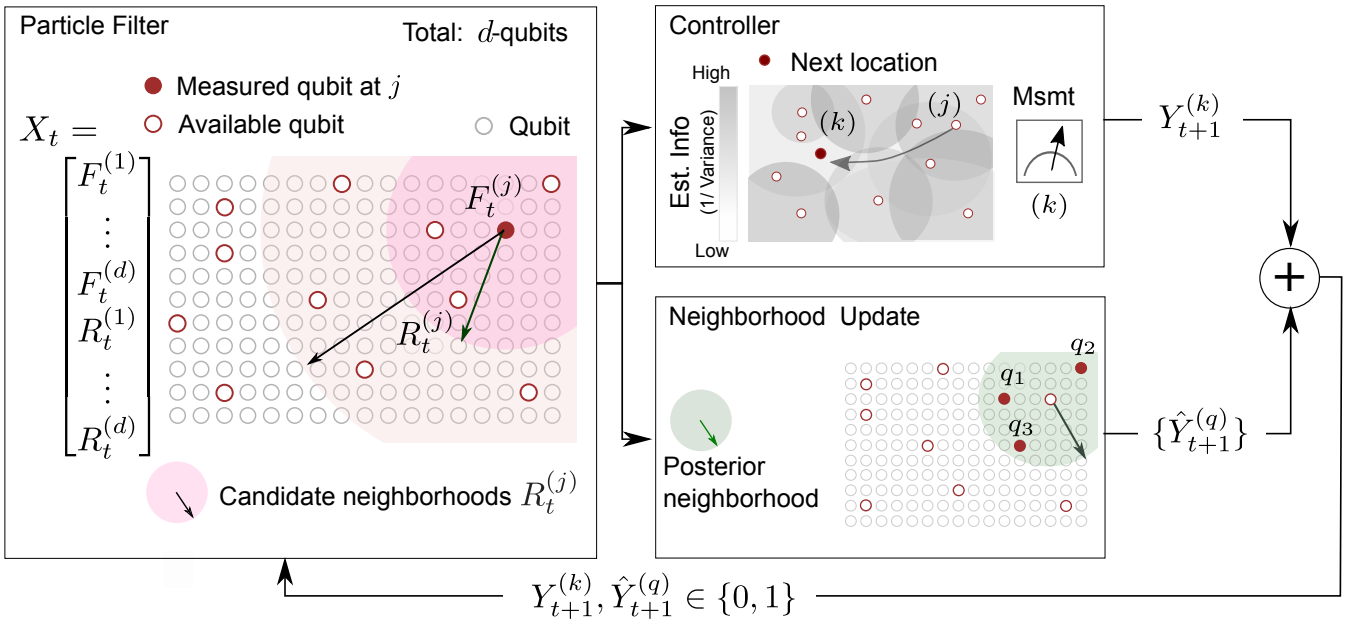

FIG. 2. Schematic overview of a QSLAM iteration. A particle filter estimates the map, F_t, and discovers neighborhoods (circular shaded) parameterized by $R_t^{(j)}$ for sharing state information about site j. Posterior state estimates from the particle filter are used by the controller, to choose site k as the location of the next physical measurement, $Y_{t+1}^{(k)}$ based on regions of highest estimated uncertainty (middle top). Meanwhile, posterior state estimates are also used to share information at j within the posterior neighborhood, Q via data messages, $\hat{Y}_{t+1}^{(q)}$, between all neighbouring qubits $q \in Q$ (middle bottom) before commencing the next iteration. The input / output information loop is the outermost loop formed by a single physical measurement (in notation, Y) and a set of data messages (in notation, $\hat{Y}$). The arrow to site k from j does not relate to the transfer of any information, but rather, adaptive measurement scheduling by the controller to maximize information utility from the next physical measurement choice k.

of particles at step t is replaced by a set of "offspring" particles, where the probability that a parent is chosen to represent itself in the next iteration (with replacement) is directly proportional to its weight. Over many iterations, only the particles with the highest weights survive and these surviving particles form the maximum-likelihood estimate of the true X_t in our algorithm.

The key operating principle of QSLAM is that we locally estimate the map value, $F_t^{(j)}$ (a qubit phase shift with value between $[0, \pi]$ radians), before globally sharing the map information at the measured qubit j with neighboring qubits $q \in Q_t$ in the vicinity of j. The algorithm is responsible for determining the appropriate size of the neighborhood, Q_t, parameterized by the length-scale value, $R_t^{(j)}$ (left panel, Fig. 2). In practice, Q_t eventually represents the set of qubits in a posterior neighborhood at j at the end of iteration t, and this set shrinks or grows as inference proceeds. We are ignorant, a priori, of any knowledge of $R_t^{(j)}$ and estimates take values between R_{min}, approximately the inter-qubit spacing, and R_{max}, the size of the qubit array, in units of distance. The collection of estimated map values and length-scales, $X_t^{(j)} := \{F_t^{(j)}, R_t^{(j)}\}$, at every qubit, $j = 1, 2, \ldots d$, is depicted as an extended state vector.

We rely on an iterative maximum likelihood procedure within each iteration of the particle filter to solve the QSLAM inference problem, as the size of the state-space generally impedes analytic solutions even in classical settings [26, 32]. In each iteration, we update F_t assuming X_{t-1} is known, and subsequently update R_t based on F_t. This structure is unique and requires two particle types; α particles carry information about the full state vector, X_t, while β particles discover the optimal information-sharing neighborhood size, $R_t^{(j)}$, around qubit j. The two different types of particle sets are then used to manipulate the joint probability distribution defined over F_t and R_t such that we numerically implement an iterative maximum likelihood procedure. The final result of the particle filtering step in Fig. 2 is a posterior estimate of X_t which represents our best knowledge given measurement data.

This estimate is now passed to an autonomous measurement scheduler, the QSLAM controller, which attempts to maximize the information utility from each measurement. The QSLAM controller adaptively selects the location k for the next physical measurement by choosing the region where posterior state variance is maximally uncertain (Fig. 2, top-middle panel), and that new measurement outcome, once collected, is denoted $Y_{t+1}^{(k)}$ (a posterior state variance is typically estimated using the properties of particles inside the filter [33]). Mean-

while, the posterior state information at step t is also shared with the set Q_t in the vicinity of qubit j (Fig. 2, bottom-middle panel). The shared information within this neighborhood is denoted by the set $\{\hat{Y}_{t+1}^{(q)}, q \in Q_{t+1}\}$ and serves as an effective state estimate that is taken as an input to the algorithm in a manner similar to a physical measurement. Jointly, the new actual physical measurement, $Y_{t+1}^{(k)}$, and the set of shared information $\{\hat{Y}_{t+1}^{(q)}\}$ form the inputs for the next iteration, $t+1$. Further technical information on the overall QSLAM implementation is presented in *Methods*.

We now provide a summary of the particle-filtering implementation used here, and highlight modifications that are unique to QSLAM. Under the conditions articulated above, the QSLAM filtering problem requires only two specifications: a prior or initial distribution for the state vector, X_0, at $t = 0$, and a likelihood function that assigns each particle with an appropriate weight based on measurement data. Assuming a uniform prior, we only need to define the global likelihood function incorporating both particle-types. We label α-particles by the set of numbers $\{1, 2, \ldots, n_\alpha\}$; for each α particle, we also associate a set of β particles denoted $\beta^{(\alpha)}$, with β taking values $\{1, 2, \ldots, n_\beta\}$. Each α particle is weighted or scored by a so-called likelihood function. This likelihood function is written in notation as $g_1(\lambda_1, Y_t^{(j)})$, where λ_1 is a parameter of the QSLAM model and $Y_t^{(j)}$ makes explicit that only one physical measurement at one location is received per iteration. A single $\beta^{(\alpha)}$-particle inherits the state from its α-parent, but additionally acquires a single, uniformly distributed sample for $R_t^{(j)}$ from the length-scale prior distribution. The β-particles are scored by a separate likelihood function, $g_2(\lambda_2, Q_t)$, where λ_2 is another parameter of the QSLAM model. Then the total likelihood for an $(\alpha, \beta^{(\alpha)})$ pair is given by the product of the α and β particle weights.

The functions $g_1(\lambda_1, Y_t^{(j)})$ and $g_2(\lambda_2, Q_t)$ are likelihood functions used to score particles inside the QSLAM particle filter, and their mathematical definitions can be found in the Supplementary Materials. These functions are derived by representing the noise affecting the physical system via probability density functions. The function $g_1(\lambda_1, Y_t^{(j)})$ describes measurement noise on a local projective qubit measurement as a truncated or a quantized Gaussian error model to specify the form of the noise density function. The function $g_2(\lambda_2, Q_t)$ represents the probability density of "true errors" arising from choosing an incorrect set of neighbourhoods while building the map. For each candidate map, the value of the field at the center is extrapolated over the neighborhood using a Gaussian kernel. Thus, what we call "true errors" are the differences between the values of the true, spatially continuous map and the QSLAM estimate, consisting of a finite set of points on a map and their associated (potentially overlapping) Gaussian neighborhoods. We assume that these errors are truncated Gaussian distributions with non-zero means and variances over many iterations.

The two free parameters, $\lambda_1, \lambda_2 \in [0, 1]$, are used to numerically tune the performance of the QSLAM particle filter. Practically, λ_1 controls how shared information is aggregated when estimating the value of the map locally and λ_2 controls how neighborhoods expand or contract in size with each iteration. Additionally, the numerically optimized values of $\lambda_1, \lambda_2 \in [0, 1]$ provide an important test of whether or not the QSLAM sharing mechanism is trivial in a given application. Non-zero values suggest that sharing information spatially actually improves performance more than just locally filtering observations for measurement noise. As $\lambda_1, \lambda_2 \to 1$, the set $\{\hat{Y}_{t+1}^{(q)}\}$, is treated as if they were the outcomes of real physical measurements by the algorithm. However, in the limit $\lambda_1, \lambda_2 \to 0$, no useful information sharing in space occurs, and QSLAM effectively reduces to the naive measurement strategy.

Detailed derivations for mathematical objects and computations associated with QSLAM, and an analysis of their properties using standard non-linear filtering theory, will be provided in a forthcoming technical manuscript.

III. RESULTS

Application of the QSLAM algorithm using both numerical simulations and real experimental data demonstrates the capabilities of this routine in arbitrary spatial arrangements of d qubits in 1D or 2D. Our evaluation procedure begins with the identification of a suitable metric for characterizing mapping performance and efficiency. We choose a Structural SIMilarity Index (SSIM) [34], frequently used to compare images in machine learning analysis. This metric compares the structural similarity between two images and is defined mathematically in *Methods*. It is found to be sensitive to improvements in the quality of images while giving robustness against *e.g.* large single-pixel errors that frequently plague norm-based metrics for vectorized images [35, 36]. In our approach, we compare the true map and its algorithmic reconstruction by calculating the SSIM; a score of zero corresponds to ideal performance, and implies that the true map and its reconstruction are identical.

We start with a challenging simulated example, in which $d = 25$ qubits are arranged in a 2D grid. The true field is partitioned into square regions with relatively low and high values (Fig. 3(a), left inset) to provide a clear structure for the mapping procedure to identify. In this case, the discontinuous change of the field values in space means that QSLAM will not be able to define a low-error candidate neighborhood for any qubits on the boundary. Both QSLAM and naive are executed over a maximum of T iterations such that $t \in [1, T]$. For simplicity, we pick values of $T \geq d$ as multiples of the number of qubits, d, such that every qubit is measured an integer number of times in the naive approach which we use as baseline in our comparison. Both QSLAM and the naive approach

terminate when $t = T$, and a single-run SSIM score is calculated using the estimated map for each algorithm in comparison with the known underlying "true" noise map. The average value of the score over 50 trials is reported as Avg. SSIM.

The Avg. SSIM score as a function of T is shown in the upper right inset of Fig. 3(a). For any choice of T, QSLAM reconstructs the true field with a lower SSIM than the naive approach, indicating better performance. The SSIM score also drops rapidly with T when using QSLAM, compared with a more gradual decline in the naive case. Increasing $T \gg d$ leads to an improvement in the performance of both algorithms but also a convergence of the scores as every qubit is likely to be measured multiple times for the largest values considered here. Representative maps for $T = 10$ and $T = 75$ measurements under both the naive and QSLAM mapping approaches are shown in Fig. 3(b-e), along with a representative adaptive measurement sequence employed in QSLAM in panel (b). In both cases, QSLAM provides estimates of the map which are closer to the true field values, whereas naive maps are dominated by estimates at the extreme ends of the range, which is characteristic for simple reconstructions using sparse measurements.

It is instructive to represent these data in a form that shows the effective performance enhancement of the QSLAM mapping procedure relative to the naive approach; the main panel in Fig. 3(a) reports the improvement in the reduction of the number of measurements required to reach a desired Avg. SSIM value. The shape of the performance improvement curve is linked to the total amount of information provided to both algorithms. At high Avg. SSIM scores on the far-right of the figure, both naive and QSLAM algorithms receive very few measurements and map-reconstructions are dominated by errors. Near the origin, an extremely large number of measurements are required to achieve low Avg. SSIM scores and the ratio of measurements between naive and QSLAM tends to unity, corresponding to the convergence mentioned above. In the intermediate regime, a broad peak indicates that QSLAM outperforms brute force measurement by up to 18× in reducing total number of qubit measurements over a range of moderate-to-low Avg. SSIM scores for map reconstruction fidelity. Similar performance is achieved for a range of other qubit-array geometries and characteristics of the underlying field (*Supplementary Materials*).

All of our results rely on appropriate tuning of the QSLAM particle filter via its parameters λ_1 and λ_2. A full numerical optimization is conducted for each T and this data is represented as solid curves in Fig. 3(a). We also demonstrate that using fixed values for these parameters only marginally degrades performance, as indicated by the dashed line in Fig. 3(a).

We now apply the QSLAM mapping algorithm to real experimental measurements on qubit arrays. In Fig. 4, we analyze Ramsey experiments conducted on an array of $^{171}Yb^{+}$ ions confined in a linear Paul trap, with trap

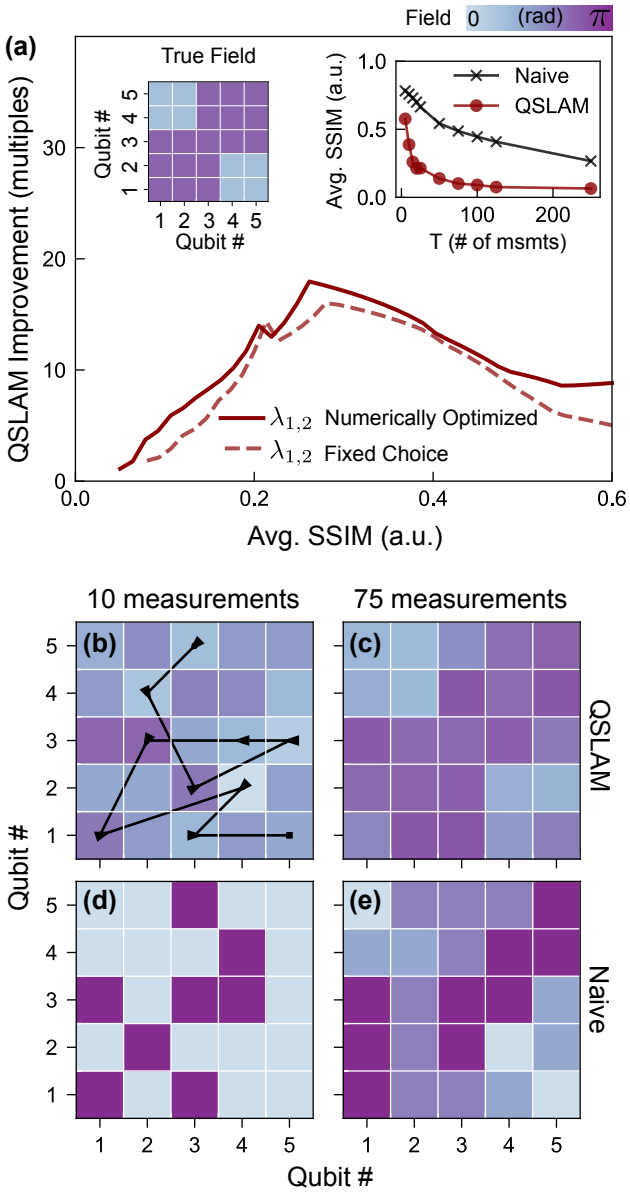

FIG. 3. (a) Upper left inset: 2D array of 25 qubits with 'square' field shown as a colorscale. Upper right inset: Avg. SSIM score vs total measurement budget T for QSLAM and naive algorithms averaged over 50 trials. Main panel: ratio of naive to QSLAM measurements vs. Avg. SSIM score for optimized λ_1, λ_2 (solid line) at each T; fixed choice $\lambda_1 = 0.95, \lambda_2 = 0.65$ (dashed line) optimized for $T = 20$. The numerical inversion of raw data in the upper right inset introduces artifacts and is unstable for small Avg. SSIM scores. (b)-(e) Columns show single-run maps using a total number of $T = 10$ or $T = 75$ measurements plotted for the QSLAM (b, d) and naive approach (c, e), with a representative control path shown in (b). The numerically optimized tuning parameters are $(\lambda_1, \lambda_2) = (0.92, 0.44)$ and $(0.93, 0.77)$ for (b),(c) respectively. The single map SSIM values are: (b) 0.64; (c) 0.13; (d) 0.80; (e) 0.47.

frequencies $\omega_{x,y,z}/2\pi \approx (1.6, 1.5, 0.5)$ MHz. Qubits are encoded in the $^2S_{1/2}$ ground-state manifold where we associate the atomic hyperfine states $|F = 0, m_F = 0\rangle$ and $|F = 1, m_F = 0\rangle$ with the qubit states $|0\rangle$ and $|1\rangle$, respectively. State initialization to $|0\rangle$ via optical pumping and state detection are performed using a laser resonant with the $^2S_{1/2}-^2P_{1/2}$ transition near 369.5 nm. Laser-induced fluorescence (corresponding to projections into state $|1\rangle$) is recorded using a spatially resolving EMCCD camera yielding simultaneous readout of all ions. In this experiment, qubit manipulation is carried out using microwave radiation at 12.6 GHz delivered via an in-vacuum antenna to all qubits globally.

The trapped ions experience a linear magnetic field gradient which leads to spatially inhomogeneous qubit frequencies over the trap volume. When manipulated using a global microwave control field, this results in a differential phase accumulation between qubits. The magnitude of the gradient is illustrated in Fig. 4(a), superimposed on an image of six ions in the bright state $|1\rangle$. We aim to probe this field gradient through the resulting qubit detuning and associated differential phase accumulation throughout each measurement repetition. A total of 25,500 Ramsey measurements with a wait time of 40 ms are performed on all six ions in parallel. For each repetition, a standard machine-learning image classification algorithm assigns a '0' (dark) or '1' (bright) to each ion based on a previously recorded set of training data. From averaging over repetitions, we construct a map of the accumulated phase (and hence the local magnetic field inducing this phase) on each qubit, shown schematically in the right inset of Fig. 4(b). We consider the field extracted from this standard averaging procedure over all repetitions of the Ramsey experiment, at each ion location, as the "true" field against which mapping estimates are compared using the SSIM score as introduced above.

We employ the full set of $6 \times 25,500$ measurements as a data-bank on which to evaluate and benchmark both algorithms. At each iteration, the algorithm determines a measurement location in the array and then randomly draws a single, discretized measurement outcome (0 or 1) for that ion from the data-bank. The rest of the algorithmic implementation proceeds as before. Accordingly, we expect that the naive approach must smoothly approach a SSIM score of zero as T increases (Fig. 4(b), left inset).

In these experiments, we experience a large measurement error arising from the necessary detection time of 750 µs associated with the relatively low quantum efficiency and effective operating speed of the EMCCD camera. This leads to an asymmetric bias due to state decays that happen in ≈ 1.5 ms ($|1\rangle \to |0\rangle$) and ≈ 30 ms ($|0\rangle \to |1\rangle$) under the laser power and quantization magnetic field strength used in our experiment. Despite this complication, we again find that QSLAM outperforms the naive mapping algorithm by a factor of $2-3$ in the number of required measurements (Fig. 4(b)), with expected behavior at the extremal values of Avg. SSIM .

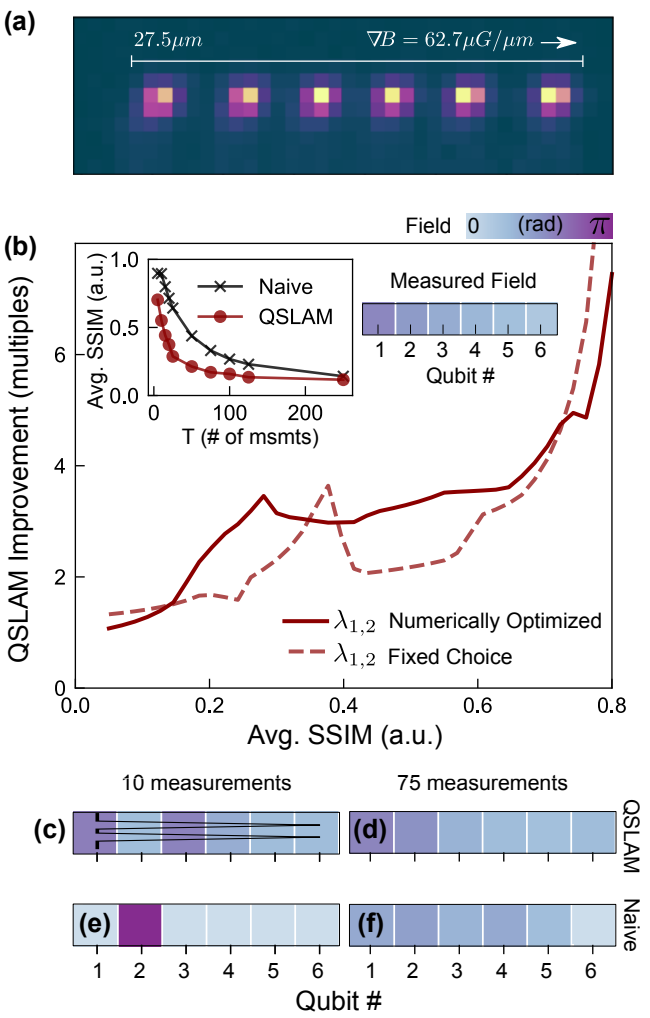

FIG. 4. (a) Image of six trapped $^{171}Yb^+$ ions in the fluorescing $|1\rangle$ state averaged over multiple exposures of 750 µs. (b) Upper right inset: Color scale indicating a phase shift induced by a magnetic field gradient across 1D array of 6 qubits. Upper left inset: Avg. SSIM score vs total measurement budget T for QSLAM and naive algorithms averaged over 50 trials. Main panel: ratio of naive to QSLAM measurements vs. Avg. SSIM score for optimized λ_1, λ_2 (solid line) at each T; fixed choice $(\lambda_1, \lambda_2) = (0.95, 0.97)$ optimized for $T = 20$. The numerical inversion of raw data in the upper right inset introduces artifacts and is unstable for small SSIM scores. (c)-(f) Columns show single-run maps using a total number of $T = 10$ or $T = 75$ measurements plotted for the QSLAM (c, d) and naive (e, f), with a representative control path shown in (c). The numerically optimized tuning parameters are $(\lambda_1, \lambda_2) = (0.99, 0.92)$ for both (c),(d). Single map SSIM values are (c) 0.46; (d) 0.05; (e) 0.90; (f) 0.34.

IV. CONCLUSION

In this work we presented QSLAM - a framework for autonomous learning where we reconstruct an unknown spatial noise field by efficiently scheduling measurements on multi-qubit devices. We developed a novel, iterative,

maximum-likelihood procedure that implements a two-layer particle filter to share state-estimation information between qubits within small spatial neighborhoods, via a mapping of the underlying spatial correlations of the noise field. An autonomous controller schedules future measurements in order to reduce the estimated uncertainty of the map reconstruction. Numerical simulations and calculations run on real experimental data demonstrated that QSLAM outperforms a naive mapping procedure by reducing the required number of measurements up to 18× in achieving a target map similarity score. Beyond these example demonstrations, the key numerical evidence for the correctness of QSLAM's functionality came from the observation that the optimized values of $\lambda_1, \lambda_2 \gg 0$ for all results reported here. Since numerically optimized values for these parameters were found to be non-zero, we conclude information sharing in QSLAM is non-trivial and the algorithm departs substantially from a brute-force measurement strategy. This contributes to the demonstrated improvements in Avg. SSIM scores when using QSLAM in Figs. 3 and 4.

The framework we have introduced is flexible and can accommodate temporal variations in the system such as drifts in the map and changes in the availability of sensor qubits. We are also excited to explore how exploitation of hardware platforms in which qubit locations are not rigidly fixed during fabrication, such as with trapped-ions, may allow sub-lattice spatial resolution by dynamically translating individual qubits during the mapping procedure. Our work is part of an exciting new area of future study exploring the intersection between hardware architectures and control solutions [37] in NISQ-era quantum computers, and may also have bearing on distributed networks of quantum sensors.

METHODS

We summarize the structure of the QSLAM algorithm in Algorithm 1 using a pseudocode representation.

The first part of the algorithm consists of an initialization procedure which ensures all particles are sampled from the prior for extended state vector at $t = 0$, giving X_0. All particles are equally weighted before measurement data is received.

For $t > 0$, the function `PropagateStates` represents the transition probability distribution for Markov X_t i.e. it represents identity dynamics and is a placeholder for future extensions to model dynamical F_t. In each t, a single physical measurement is received. This triggers a set of update rules for F_t and R_t. We note that the state variables, F_t and R_t, are updated in a specific order within each time-step t. The order of these computations correspond to the iterative maximum likelihood approximation for the QSLAM framework.

The form of the update rules reflect QSLAM's unique state vector and QSLAM's information sharing procedures. Map and neighborhood information is carried via different particle types, and information sharing is implemented via particle updates and re-sampling steps. For each type of particle, the weights are computed according to QSLAM likelihood functions $g_1(\lambda_1, Y_t^{(j)})$ and $g_2(\lambda_2, Q_t)$.

Algorithm 1 QSLAM

procedure QSLAM(d qubit locations, λ_1, λ_2)

  if $t = 0$ **then**
    procedure INITIALIZE(X_0)
      for $\alpha \in \{1, 2, \ldots, n_\alpha\}$ **do**
        Initially sample $x_0^{(\alpha)} \sim \pi_0$
        Initially compute $W_0^{(\alpha)} = \frac{1}{n_\alpha}$
      end for
    end procedure
  end if

  while $1 \leq t < T$ **do**
    if Controller **then**
      $j_t, Y_t^{(j_t)} \leftarrow$ CONTROLLER(X_{t-1}) ▷ Qubit j, at t
    end if
    for $\alpha \in \{1, 2, \ldots, n_\alpha\}$ **do**
      $\{x_t^{(\alpha)}\} \leftarrow$ PROPAGATESTATES($\{x_{t-1}^{(\alpha)}\}$)
      Update $F_t^{(\cdot),(\alpha)}$ via $\{Y_t^{(j_t)}, \{\hat{Y}_t^{(q_t)}\}, \lambda_1\}$
      $\{\{x_t, W_t\}^{(\alpha,\beta_\alpha)}\} \leftarrow$ COMPUTEWEIGHTS($\{x_t^{(\alpha)}\}$)
      $\{x_t^{(\alpha,\beta_\alpha)}, \frac{1}{n_\alpha n_\beta}\} \leftarrow$ RESAMPLE($\{\{x_t, W_t\}^{(\alpha,\beta_\alpha)}\}$)
      Update $R_t^{(j_t),(\alpha)}$
      $\{\{x_t, W_t\}^{(\alpha)}\} \leftarrow$ COLLAPSEβ($\{x_t^{(\alpha,\beta_\alpha)}, \frac{1}{n_\alpha n_\beta}\}$)
      $\{x_t^{(\alpha)}, \frac{1}{n_\alpha}\} \leftarrow$ RESAMPLE($\{\{x_t, W_t\}^{(\alpha)}\}$)
    end for
    $\{\hat{Y}_{t+1}^{(q)}\}_{q \in Q_{t+1}} \leftarrow$ GENERATE$\hat{Y}$(Posterior X_t)
  end while
end procedure

function COMPUTEWEIGHTS($\{x_t^\alpha\}$)
  for $\alpha \in \{1, 2, \ldots, n_\alpha\}$ **do**
    Compute $\tilde{W}_t^{(\alpha)} = g_1(\lambda_1, Y_t^{(j)})$
    $\{x_t^{(\alpha,\beta_\alpha)}\} \leftarrow$ Generate β-layer
    for $\beta_\alpha \in \{1, 2, \ldots, n_\beta\}$ **do**
      Compute $\tilde{W}_t^{(\beta_\alpha|\alpha)} = g_2(\lambda_2, Q_t)$
    end for
    Normalize $\tilde{W}_t^{(\beta_\alpha|\alpha)}$
  end for
  Normalize $\tilde{W}_t^{(\alpha)}$
  Compute $W_t^{(\alpha,\beta_\alpha)} = \tilde{W}_t^{(\beta_\alpha|\alpha)} \tilde{W}_t^{(\alpha)} \quad \forall\{\alpha, \{\beta_\alpha\}\}$
  Return $n_\alpha n_\beta$ particles and weights $\{\{x_t, W_t\}^{(\alpha,\beta_\alpha)}\}$
end function

Despite atypical computational steps and unique likelihood functions, the total number of particles is constant at the beginning and end of each time-step t and the particle branching mechanism for QSLAM remains a multinomial branching process. These multi-nomial branching processes are characteristic of standard particle filtering techniques [33].

A. Structural Similarity Metric Definition

For all analysis, we use a risk metric called the Structural Similarity Index (SSIM) [34]. This metric is used to conduct optimization of QSLAM parameters and assess performance relative to the naive measurement strategy. For two vectorized images x and y, the metric is defined as:

$$s(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (1)$$

$$\text{SSIM}(x,y) := |1 - s(x,y)| \quad (2)$$

In the formula above, $\mu_i, \sigma_i^2, i = x, y$ represent the sample estimates of the means and variances of the respective vectorized images, and σ_{xy} captures correlation between images. The term $s(x,y)$ is the key metric developed in [34] and it includes arbitrary constants $C_1 = C_2 = 0.01$ which stabilize the metric for images with means or variances close to zero. The ideal score given by $s(x,y)$ is unity, and corresponds uniquely to the case $x = y$. We report the absolute value of the deviations from the ideal score of unity, where the direction of the deviation is ignored as given by $\text{SSIM}(x,y)$. For our application, this $\text{SSIM}(x,y)$ metric lies between $[0,1]$ (negative values of $s(x,y)$ are not seen in our numerical demonstrations). We report the average of $\text{SSIM}(x,y)$ values over 50 trials as Avg. SSIM.

ACKNOWLEDGMENTS

Authors thank V.M. Frey for proposing methods for single-ion state detection using camera images, and S. Sukkarieh, A.C. Doherty, and M. Hush for useful discussions. This work partially supported by the ARC Centre of Excellence for Engineered Quantum Systems CE170100009, the US Army Research Office under Contract W911NF-12-R-0012, and a private grant from H. & A. Harley.

DATA AVAILABILITY

All simulated and experimental data to reproduce all figures can be accessed via links in the Supplementary Materials without restrictions.

CODE AVAILABILITY

Minimally reproducing equations for QSLAM and links to the code-base are provided in the Supplementary Materials without restrictions.

AUTHOR CONTRIBUTIONS

The QSLAM theoretical framework and numerical implementations were devised by R. Gupta based on research directions set by M.J. Biercuk. R. Gupta and M.J. Biercuk co-wrote the paper. A. Milne, C. Edmunds and C. Hempel led all experimental efforts and contributed to the paper draft.

COMPETING INTERESTS

The authors declare no competing financial interests.

V. MATERIALS AND CORRESPONDENCE

Correspondence to R. Gupta.

[1] John Preskill, "Quantum Computing in the NISQ era and beyond," Quantum **2**, 79 (2018).
[2] Norman Y Yao, Liang Jiang, Alexey V Gorshkov, Peter C Maurer, Geza Giedke, J Ignacio Cirac, and Mikhail D Lukin, "Scalable architecture for a room temperature solid-state quantum information processor," Nature communications **3**, 800 (2012).
[3] C Monroe, R Raussendorf, A Ruthven, KR Brown, P Maunz, L-M Duan, and J Kim, "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," Physical Review A **89**, 022317 (2014).
[4] M Veldhorst, HGJ Eenink, CH Yang, and AS Dzurak, "Silicon CMOS architecture for a spin-based quantum computer," Nature communications **8**, 1766 (2017).
[5] N Cody Jones, Rodney Van Meter, Austin G Fowler, Peter L McMahon, Jungsang Kim, Thaddeus D Ladd, and Yoshihisa Yamamoto, "Layered architecture for quantum computing," Physical Review X **2**, 031007 (2012).
[6] David Kielpinski, Chris Monroe, and David J Wineland, "Architecture for a large-scale ion-trap quantum computer," Nature **417**, 709 (2002).
[7] David P Franke, James S Clarke, Lieven MK Vandersypen, and Menno Veldhorst, "Rents rule and extensibility in quantum computing," Microprocessors and Microsystems (2019).
[8] A. Cooper, E. Magesan, H. N. Yum, and P. Cappellaro, "Time-resolved magnetic sensing with electronic spins in diamond," Nature Communications **5**, 3141 EP – (2014).
[9] Masashi Hirose and Paola Cappellaro, "Coherent feedback control of a single qubit in diamond," Nature **532**, 77 EP – (2016).
[10] Kenneth R Brown, Jungsang Kim, and Christopher Monroe, "Co-designing a scalable quantum computer

with trapped atomic ions," NPJ Quantum Information **2**, 16034 EP – (2016).

[11] Sandeep Mavadia, Virginia Frey, Jarrah Sastrawan, Stephen Dona, and Michael J. Biercuk, "Prediction and real-time compensation of qubit decoherence via machine learning," Nature Communications **8**, 14106 EP – (2017).

[12] Riddhi Swaroop Gupta and Michael J Biercuk, "Machine learning for predictive estimation of qubit dynamics subject to dephasing," Physical Review Applied **9**, 064042 (2018).

[13] Lukas Postler, Ángel Rivas, Philipp Schindler, Alexander Erhard, Roman Stricker, Daniel Nigg, Thomas Monz, Rainer Blatt, and Markus Müller, "Experimental quantification of spatial correlations in quantum dynamics," arXiv preprint arXiv:1806.08088 (2018).

[14] Davide Venturelli, Minh Do, Eleanor Rieffel, and Jeremy Frank, "Compiling quantum circuits to realistic hardware architectures using temporal planners," Quantum Science and Technology **3**, 025004 (2018).

[15] Prakash Murali, Jonathan M Baker, Ali Javadi Abhari, Frederic T Chong, and Margaret Martonosi, "Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers," arXiv preprint arXiv:1901.11054 (2019).

[16] Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I Schuster, Henry Hoffman, and Fred T Chong, "Optimized compilation of aggregated instructions for realistic quantum computers," arXiv preprint arXiv:1902.01474 (2019).

[17] Davide Venturelli, Minh Do, Kyle Booth, Eleanor Rieffel, Jeremy Frank, and Christopher Beck, "Optimization and planning approaches for low-level hardware compilation of quantum circuits," in *APS Meeting Abstracts* (2018).

[18] Swamit S Tannu and Moinuddin K Qureshi, "A case for variability-aware policies for NISQ-era quantum computers," arXiv preprint arXiv:1805.10224 (2018).

[19] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," IEEE Transactions on robotics **32**, 1309–1332 (2016).

[20] Niclas Bergman, "Recursive bayesian estimation," Department of Electrical Engineering, Linköping University, Linköping Studies in Science and Technology. Doctoral dissertation **579**, 11 (1999).

[21] Cyrill Stachniss, Wolfram Burgard, *et al.*, "Particle filters for robot navigation," Foundations and Trends® in Robotics **3**, 211–282 (2014).

[22] Hugh Durrant-Whyte and Tim Bailey, "Simultaneous localization and mapping: Part I, robotics & automation magazine," IEEE **13**, 99–110 (2006).

[23] Tim Bailey and Hugh Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," IEEE Robotics & Automation Magazine **13**, 108–117 (2006).

[24] Kevin P Murphy, "Bayesian map learning in dynamic environments," in *Advances in Neural Information Processing Systems* (2000) pp. 1015–1021.

[25] Andrew Howard, "Multi-robot simultaneous localization and mapping using particle filters," The International Journal of Robotics Research **25**, 1243–1256 (2006).

[26] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic robotics* (MIT press, 2005).

[27] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," Autonomous Robots **5**, 253–271 (1998).

[28] Arnaud Doucet, Nando De Freitas, and Neil Gordon, *Sequential Monte Carlo methods in practice* (Springer, 2001) pp. 3–14.

[29] Kristopher R Beevers and Wesley H Huang, "Fixed-lag sampling strategies for particle filtering SLAM," in *Proceedings 2007 IEEE International Conference on Robotics and Automation* (IEEE, 2007) pp. 2433–2438.

[30] Giorgio Grisettiyz, Cyrill Stachniss, and Wolfram Burgard, "Improving grid-based slam with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (IEEE, 2005) pp. 2432–2437.

[31] Jonathan Poterjoy, "A localized particle filter for high-dimensional nonlinear systems," Monthly Weather Review **144**, 59–76 (2016).

[32] Sebastian Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," The International Journal of Robotics Research **20**, 335–363 (2001).

[33] Alan Bain and Dan Crisan, *Fundamentals of Stochastic Filtering*, Stochastic Modelling and Applied Probability (Springer, 2009).

[34] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE transactions on image processing **13**, 600–612 (2004).

[35] Yihua Chen, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti, "Similarity-based classification: Concepts and algorithms," Journal of Machine Learning Research **10**, 747–776 (2009).

[36] Zhou Wang and Alan C Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," IEEE signal processing magazine **26**, 98–117 (2009).

[37] Harrison Ball, Trung Nguyen, Philip H. W. Leong, and Michael J. Biercuk, "Functional basis for efficient physical layer classical control in quantum processors," Phys. Rev. Applied **6**, 064009 (2016).

Supplementary Materials

The Supplementary Materials provide additional performance results to support the main findings of the paper. We also provide the minimal set of equations required to implement QSLAM as a non-linear, particle filter. These equations specify the computational steps to accompany the *Methods* section of the main text.

I. SUPPORTING RESULTS

In this section, we provide details about the optimization approach used to produce all data-figures. We also report additional simulation results for various underlying spatial fields and qubit configurations.

A. (λ_1, λ_2) Optimization

The numerical optimization of QSLAM parameters, (λ_1, λ_2), helps assess whether the QSLAM algorithm provides meaningful adaptive measurement scheduling, relative to a brute force measurement strategy, for any specific practical application.

In Fig. 1, we plot the Avg. SSIM score for 50 randomly sampled (λ_1, λ_2) pairs and we compare this with the expected value of the Avg. SSIM score for $\lambda_1, \lambda_2 = 0$ in Fig. 1. The rows of Fig. 1 correspond to three different simulations of the true field: a 1D step field of Fig. 2, a 2D square field reported in the main text as Fig. 3, and a 2D Gaussian field corresponding to Fig. 3. In each panel, a black dot represents a randomly selected (λ_1, λ_2) pair. The shaded circle selects a pair with an Avg. SSIM score better than $\lambda_1, \lambda_2 = 0$ by at least 0.1 in arbitrary units (10% of maximal expected deviation). The color of the shaded circle refers to the actual value of the Avg. SSIM. The crimson star indicates the lowest score achieved over all pairs, and defines the numerically optimized (λ_1, λ_2) values used for analysis. This procedure is repeated for all T in the solid lines of Figs. 2 and 3 and Fig. 3 of main text. The fixed choice case refers to finding optimal (λ_1, λ_2) pair for $T = 20$ and using this pair for all $T \neq 20$. To aid visual comparisons, we use the same set of random pairs (λ_1, λ_2) across all experiments, but results hold if candidate parameters are randomized across experiments. For the numerical demonstrations in the main text and the Supplement, it is seen that numerically optimized $\lambda_1, \lambda_2 \neq 0$ and QSLAM sharing is non-trivially contributing to the overall inference procedure.

Of the experiments reported in Fig. 1, one observes that performance improves from bottom left to top right corners of all panels, indicating a performance improvement away from zero values for both parameters. With $\lambda_1, \lambda_2 \equiv 1$, shared information in QSLAM is treated on an equal footing with information obtained from physical data. For $\lambda_1, \lambda_2 \equiv 0$, QSLAM effectively reduces to a brute force measurement strategy but with uniformly randomly sampled locations for performing measurements. Since performance appears to improve as the two parameters, λ_1, λ_2 move away from zero and this trend is unchanged even as the total amount of measurement data increases (left column to right), we interpret this as numerical evidence that the sharing mechanism in QSLAM framework is both non-trivial and correct in the large data limit.

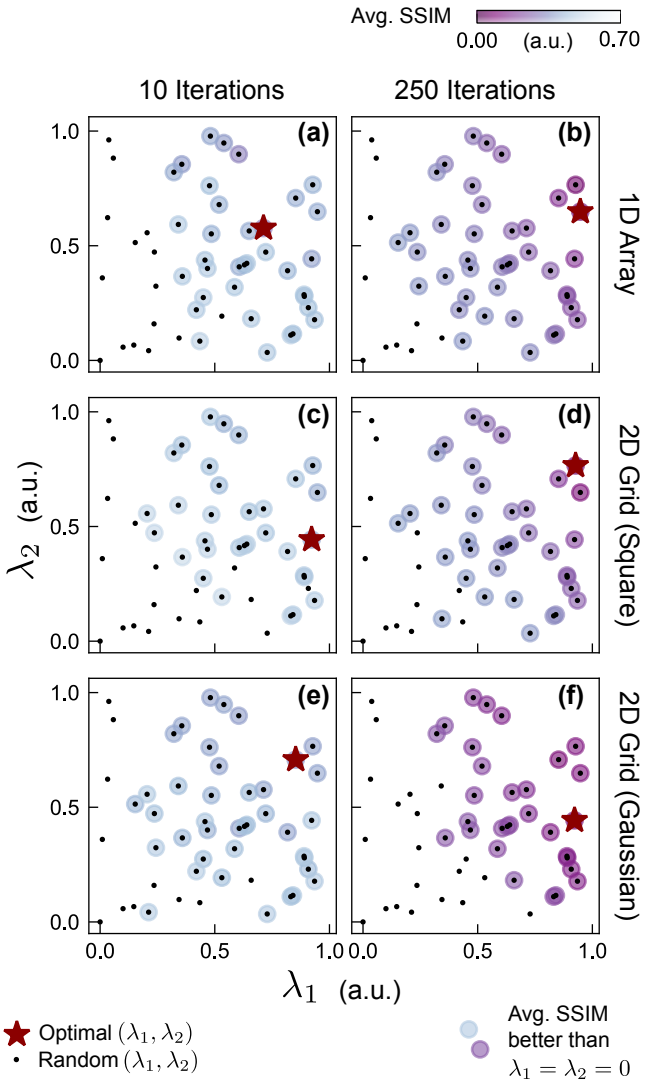

FIG. 1. Numerical optimization of λ_1, λ_2 parameters. Rows correspond to different spatial field and qubit array configurations; columns correspond to $T = 10, 250$. Each panel depicts 50 randomly chosen (λ_1, λ_2) pairs (black dots). Shaded blue-violet circles highlight (λ_1, λ_2) pairs with a substantially lower Avg. SSIM score compared with $\lambda_1 = \lambda_2 = 0$ case; color-scale conveys actual Avg. SSIM value. Optimal (λ_1, λ_2) pair corresponds to lowest Avg. SSIM score (crimson star).

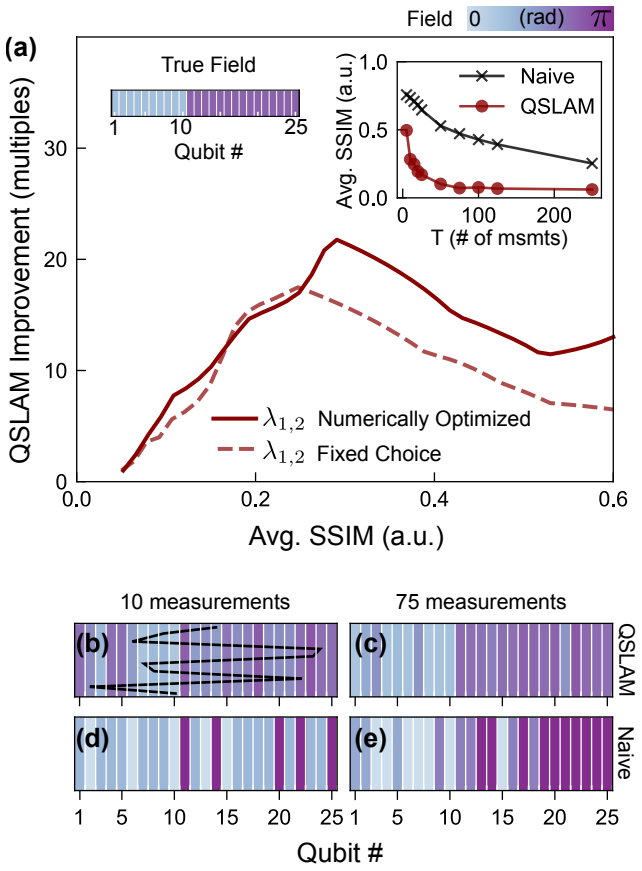

FIG. 2. (a) Upper left inset: 1D array of 25 qubits in 'step' field shown as a colorscale. Upper right inset: Avg. SSIM score vs total measurement budget T for QSLAM and naive algorithms averaged over 50 trials. Main panel: ratio of naive to QSLAM measurements vs. Avg. SSIM score for optimized λ_1, λ_2 (solid line) at each T; fixed choice $\lambda_1 = 0.93, \lambda_2 = 0.77$ (dashed line) optimized for $T = 20$. The numerical inversion of raw data in the upper right inset introduces artifacts and is unstable for small SSIM scores. (b)-(e) Columns show single-run maps using a total number of $T = 10$ or $T = 75$ measurements plotted for the QSLAM (b, d) and naive approach (c, e), with a representative control path shown in (b). The numerically optimized tuning parameters are $(\lambda_1, \lambda_2) = (0.72, 0.58)$ and $(0.95, 0.65)$ for (b),(c) respectively. The single map SSIM values are: (b) 0.26; (c) 0.08; (d) 0.70; (e) 0.54.

B. Additional Numerical Simulations

The performance of QSLAM is of course conditioned on the spatial variation in the true field relative to the configuration of the qubit-grid.

Fig. 2 provides a simulation of the performance of QSLAM when neighbours are constrained to a line in 1D. This 1D case can be compared with Fig. 3 of the main text, where we re-stack 25 qubits from 2D to 1D while keeping the overall ratio of qubits subject to low and high field regions the same. By reducing spatial dimensions on the qubit-grid, we influence neighbourhood selection within the algorithm.

In Fig. 3, we provide an example of a true field where the value of the field changes by almost every 'pixel' that is, spatial variation of the true field is 'fast' compared to results in Fig. 3 of main text. From standard classical sampling considerations, we expect that the spatial variation in the true field relative to the inter-qubit spacing in any dimension will affect overall performance.

In these additional simulations, QSLAM still outperforms naive by 2×-15× for a large range of error scores and the ratio between QSLAM and naive measurements approaches unity in the large data limit.

II. MINIMALLY REPRODUCING EQUATIONS FOR QSLAM FILTERING PROBLEM

This section introduces mathematical details for the QSLAM non-linear filter. We provide definitions, equations and the order of computations required to reproduce the algorithm. A detailed introduction to these mathematical objects and computations; supporting derivations, and an analysis of their properties using standard non-linear filtering theory, will be provided in a forthcoming technical manuscript.

A. Physical Measurement Model

Our system consists of d qubits and a total number of T measurements are taken at $t = 1, 2, \ldots, T$. Qubit locations are labeled by $\{1, 2, \ldots d\}$. A measurement $Y_t^{(j_t)} \in \{0, 1\}$ is a random variable obtained by measuring a single qubit at the location $j_t \in \{1, 2, \ldots d\}$ at iteration t.

The set of phases for all qubits in the system is the random vector F_t and it is referred to as the 'true map' or equivalently, as the set of 'map values'. This state vector $F_t \in \mathbb{S}_F := [0, \pi]^d$ takes in values between $[0, \pi]$ and the map-value, $F_t^{(j_t)}$ is a noise-induced Born phase for a qubit at location j_t. Here, and elsewhere, the notation $\mathbb{S}$ refers to a complete, separable metric space and $\mathcal{S} \equiv \sigma(\mathbb{S})$ is the associated Borel σ-algebra generated by $\mathbb{S}$ for a particular random variable, in this case, for F. These definitions are required to specify spaces of probability measures and continuous, bounded, Borel-measurable functions for particle filtering methods.

The measurement model for our state vector incorporates a projective single-qubit measurement. For a given location j_t and iteration t, the measurement model is:

$$Y_t^{(j_t)} := \mathcal{Q}(\frac{1}{2}\cos(F_t^{(j_t)}) + v_t + \frac{1}{2}) \tag{1}$$

Here, $\mathcal{Q}$ represents a Bernoulli trial parameterized by a probability that depends on the map-value $F_t^{(j_t)}$ and a noise term, v_t.

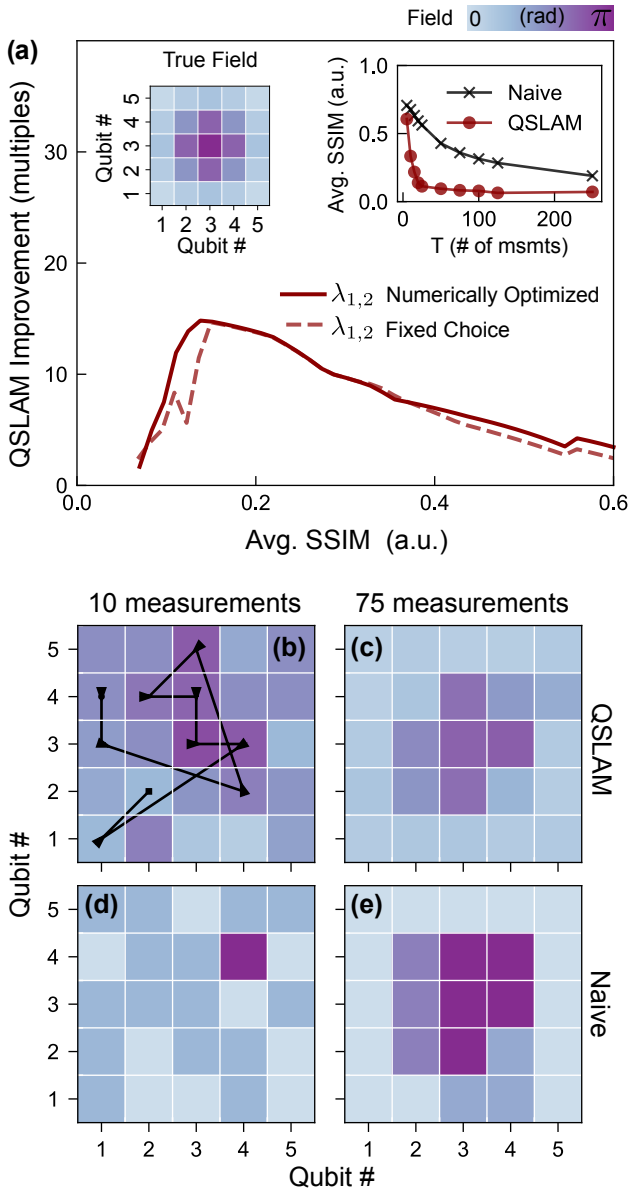

FIG. 3. (a) Upper left inset: 2D array of 25 qubits in 5×5 'Gaussian' field shown as a colorscale. Upper right inset: Avg. SSIM score vs total measurement budget T for QSLAM and naive algorithms averaged over 50 trials. Main panel: ratio of naive to QSLAM measurements vs. Avg. SSIM score for optimized λ_1, λ_2 (solid line) at each T; fixed choice $\lambda_1 = 0.85, \lambda_2 = 0.71$ (dashed line) optimized for $T = 20$. The numerical inversion of raw data in the upper right inset introduces artifacts and is unstable for small SSIM scores. (b)-(e) Columns show single-run maps using a total number of $T = 10$ or $T = 75$ measurements plotted for the QSLAM (b, d) and naive approach (c, e), with a representative control path shown in (b). The numerically optimized tuning parameters are $(\lambda_1, \lambda_2) = (0.85, 0.71)$ and $(0.92, 0.44)$ for (b),(c) respectively. The single map SSIM values are: (b) 0.24; (c) 0.16; (d) 0.74; (e) 0.38.

This measurement noise term has a form resembling a truncated Gaussian error model with zero mean and Σ_v variance; but it is derived from a study of amplitude-quantised sensor error in classical signal processing. The measurement model leads to a likelihood function given by:

$$g_1(\lambda_1, Y_t^{(j_t)}) := \frac{\rho_0}{2} + \frac{\rho_0 \cos(F_t^{(j_t)})}{2}\left(\delta(Y_t^{(j_t)} - 1) - \delta(Y_t^{(j_t)})\right) \tag{2}$$

$$\rho_0 := \mathrm{erf}\left(\frac{2b}{\sqrt{2\Sigma_v}}\right) + \frac{\sqrt{2\Sigma_v}}{2b}\frac{e^{-(\frac{2b}{\sqrt{2\Sigma_v}})^2}}{\sqrt{\pi}} - \frac{1}{2b}\frac{\sqrt{2\Sigma_v}}{\sqrt{\pi}} \tag{3}$$

$$b := 1/2 \tag{4}$$

In the likelihood function above, the parameters ρ_0, b are set by the properties of measurement noise when sensor data is allowed to take on only discrete binary values. In practice, the term $F_t^{(j_t)}$ depends on λ_1 under certain particle-filtering approximations, as introduced in the final section. The prior for F_0 is uniformly distributed.

B. Information Sharing Mechanism

Next, we introduce the mechanism to share state information between qubits within a local neighbourhood. We use the concept of 'blurring' state information from a point value at j_t into a small neighbourhood around j_t, and the set of neighbouring qubits in this region is given by Q_t. Both the size of the neighbourhood and the interpolation of map values within the neighborhood is parameterized by a random length-scale variable, $R_t^{(j_t)}$, at j_t. For some $R_{min} \in \mathbb{R} > 0$, we define the state vector $R_t \in \mathbb{S}_R := [R_{min}, R_{max}]^d$ the set of random variables associated with d qubit locations, taking in values between $[R_{min}, R_{max}]$, such that $R_t^{(j_t)}$ is the j_t element of R_t. The length-scale parameterizes a Gaussian function with amplitude $F_t^{(j_t)}$ that spreads the point-value of the phase $F_t^{(j_t)}$, at j_t, into a small neighbourhood. Let $\nu_{(j_t,q)}$ be the separation distance between the qubit at j_t and a point q in a neighbourhood around j_t. Then the smeared phase value is given by:

$$\bar{F}(\nu_{(j_t,q)}) := F_t^{(j_t)} \exp\left(\frac{-\nu_{(j_t,q)}^2}{(R_t^{(j_t)})^2}\right) \tag{5}$$

The point value of the phase is regained in the limit $R_t^{(j_t)} \to 0$, and the smallest spatial resolution sets the value of R_{min}.

The neighbourhood, Q_t of the measured qubit at j_t with length-scale $R_t^{(j_t)}$ is the set:

$$Q_t := \{q_t | \nu_{(j_t,q_t)} \leq k_0 R_t^{(j_t)} \quad \forall q_t \in \{1, 2, \ldots d\} \setminus \{j_t\}\}, \quad k_0 \in [1, \infty). \tag{6}$$

That is, neighbours are the set of qubits whose separations distances fall within a radius of $k_0 R_t^{(j_t)}$ about the location j_t. Here, k_0 is an arbitrary constant that truncates the neighbourhood when smeared phase values at the boundary of the neighborhood are dissimilar to those at the center. A conservative approach is to set $k_0 = 1$. The index q_t denotes the location labels for all qubits excluding the measured qubit, $q_t \in \{1, 2, \dots d\} \setminus \{j_t\}$.

Given a map F_t, we associate a likelihood function for the length-scale at j_t, $R_t^{(j_t)}$, that scores the most appropriate length-scale for sharing information between qubits. For a constant $\lambda_2 \in [0, 1]$, and a non-negative natural number τ_{q_t}, the likelihood for the length-scale at j_t compares the estimate $\mathcal{X}_{q_t}$ to the best available phase information on each neighbour:

$$g_2(\lambda_2, Q_t) := \prod_{q_t \in Q_t} \frac{1}{k_1 \sqrt{2\pi\Sigma_F}} \exp\left(-\frac{(F_t^{(q_t)} - \mathcal{X}_{q_t} - \mu_F)^2}{2\Sigma_F} \right) \tag{7}$$

$$F_t^{(q_t)} = \mathcal{X}_{q_t} + w_t, \quad \forall q_t \in Q_t \tag{8}$$

$$\mathcal{X}_{q_t} := (1 - \lambda_2^{\tau_{q_t}}) F_t^{(q_t)} + \lambda_2^{\tau_{q_t}} \bar{F}(\nu_{(j_t, q_t)}) \tag{9}$$

$$w_t \sim \mathcal{N}(\mu_F, \Sigma_F) \tag{10}$$

$$k_1 := \frac{1}{2} \left(\mathrm{erf}(\frac{\pi + \mu_F}{\sqrt{2\Sigma_F}}) + \mathrm{erf}(\frac{\pi - \mu_F}{\sqrt{2\Sigma_F}}) \right) \tag{11}$$

The quantity $\mathcal{X}_{q_t}$ is based on a weighted average of both current map estimate on the neighbouring qubit and state estimates of physically measured qubit at j_t, mediated by a factor $\lambda_2^{\tau_{q_t}}$. The noise parameters μ_F, Σ_F represent the true error in approximating a continuously varying spatial field with overlapping Gaussian functions. This error is Gaussian distributed and error values lie in the finite interval $[-\pi, \pi]$, giving rise to the constant k_1. The non-negative λ_2 is a parameter for the filtering problem, and the constant $\tau_{q_t} \leq T$ is the tally of the total number of times q_t has been physically measured, as defined in the final section. The prior for R_t is uniformly distributed.

With the definitions above, posterior state information for map-values and length-scales can be defined under a particle-filtering framework. We now generate binary data messages on the basis of posterior state information at t, which influences the inference procedure at $t + 1$. For a posterior state estimate F_t and $R_t^{(j)}$ at iteration t, a physical measurement at j_t is said to *induce* information exchange in the posterior neighbourhood Q_t (equivalently, prior at the start of $t + 1$ step before any information updates commence). In notation, these data messages are the random variables $\hat{Y}_{t+1}^{(q_{t+1})} \in \{0, 1\}$ generated using the model:

$$\hat{Y}_{t+1}^{(q_{t+1})} := \mathcal{Q}(\frac{1}{2}\cos(\mathcal{X}_{q_t}) + \frac{1}{2}) \tag{12}$$

The shared information uses a weighted average of both posterior map estimate on the neighbouring qubit and posterior state information for the physically measured qubit at j_t, mediated by a factor $\lambda_2^{\tau_{q_t}}$. Note that $\mathcal{X}_{q_t}$ is to be interpreted as the posterior Eq. (9), where $q_{t+1} \in Q_{t+1}$ (above) equals the Q_t set by posterior state estimate $R_t^{(j)}$ and the posterior F_t is used for all calculations.

C. Nonlinear Filtering

We are now in a position to state the non-linear filtering problem for QSLAM in terms of a state variable X and an observation vector Z.

We define a Markov state vector at t by $X_t = (F_t, R_t) \in \mathbb{S}_X := \mathbb{S}_F \times \mathbb{S}_R$. The filteration generated by the process X is $\mathcal{F}_t := \sigma(\{X_s, s \in [0, t]\})$ and we denote $X_{0:t} := (X_0, \dots, X_t)$ as the set of state variables until t. The observation, $Z_t = (Y_t^{(j_t)}, \{\hat{Y}_t^{(q_t)}\}_{q_t \in Q_t})$ at t is defined using a single physical measurement at j_t and a set of messages $\{\hat{Y}_t^{(q_t)}\}_{q_t \in Q_t}$ for the posterior neighbourhood Q_{t-1} about j_{t-1}. The full observational vector is $Z := \{Z_t, t = 1, 2, \dots\}$ and $Z_{0:t} := (Z_0, \dots, Z_t)$ denotes the measurement record until t.

With these definitions, we can define spaces of probability measures, where each probability measure is a function defined over the space of all possible outcomes for both X and Z, satisfying properties associated with the assignment of probabilities. These spaces of probability measures can consist of *random* measures in particle filtering techniques, from which particle filtering methods must converge to the true measure as total data and the total number of particles increase.

The filtering problem is then to compute the specific random measure π_t that is the conditional probability of X given the σ-field generated by the observation process $Z_{0:T}$:

$$\pi_t := \mathbb{P}[X_t \in A | \sigma(Z_{0:T})], \quad \forall A \in \mathcal{S}_X \tag{13}$$

$$\pi_t f = \mathbb{E}[f(X_t) | \sigma(Z_{0:T})] \quad \forall f \in B(\mathbb{S}_X), A \in \mathcal{S}_X \tag{14}$$

$$\pi_0 \sim \mathcal{U}(\mathbb{S}_X) \tag{15}$$

In the above, π_t is identified with the posterior distribution in Bayesian analysis. The appropriate state space is given by $\mathbb{S}_X$ and $\mathcal{S}_X$ is the σ-algebra generated by $\mathbb{S}_X$. Similar comments apply to the observation vector. The term $B(\mathbb{S}_X)$ refers to a space of bounded, $\mathcal{S}_X$-measurable functions which correspond to transformations of the state (e.g. dynamical evolution, measurement models) in the inference procedure.

For the class of particle methods used in our analysis, the transformation of the hidden state X to the observations Z at any t is expressed primarily via the likelihood function. In our case, the likelihood function incorporates both the physical measurement model and the information sharing mechanism. For some constants $\lambda_{1,2} \in [0, 1]$, total likelihood function for each $Z_t = z_t$ is given by:

$$g_t^{z_t} := g_1(\lambda_1, Y_t^{(j_t)}) g_2(\lambda_2, Q_t) \tag{16}$$

The measurement and map-building noise parameters that govern the level of uncertainty in the system are: $\Sigma_v, \mu_F, \Sigma_F$, where Σ_v captures variance of zero mean measurement noise from a quantized sensor Gaussian error model, and μ_F, Σ_F represent the true error in approximating a continuously varying fields with overlapping Gaussian neighborhoods. The non-negative $\lambda_1, \lambda_2 \in [0, 1]$ regulate the extent to which shared information is utilized by the inference procedure.

D. Particle Approximations and Iterative Maximum Likelihood

The filtering problem, so defined, is challenging as the space of maps and lengthscales is very large. Instead, we use an iterative maximum likelihood approach. This approach assumes that we have access to two data association mechanisms, h_1 and h_2, to update state variables at t directly using incoming data if the previous state is known. In practical cases, one only has access to estimates of states, and we use h_1 and h_2 to update the state estimates of F_t and R_t iteratively within each t. To implement these iterative maximum likelihood calculations numerically, we require two different particle sets within our filter. The computational details of these procedures are provide below and they constitute as the iterative maximum likelihood approximation for QSLAM.

For $t > 0$, the particle filter has n_α number of particles at the start and the end of an iteration. The set of particles at the beginning and end of each t are called α-type particles, indexed by the set of numbers $\{1, 2, \ldots, n_\alpha\}$. For each α-particle, we associate a set of $\beta^{(\alpha)}$-particles labeled from $\{1, 2, \ldots, n_\beta\}$. We refer to this as the 'β-layer' for an α-particle. A single $\beta^{(\alpha)}$-particle inherits the state $X_t \setminus \{R_t^{(j_t)}\}$ from its α-parent; and additionally, inherits a single uniformly distributed sample for $R_t^{(j_t)}$ from the length-scale prior distribution. The total likelihood $g_t^{z_t}$ over the entire particle set is given by the product of the α and β particle weights. These conditions yield the following update steps at t:

1. Assume X_{t-1} is known.
   (a) Update $F_t^{(j_t)}$ for each α-particle using data via h_1.
   (b) Weight each α-particle according to $g_1(\lambda_1, Y_t^{(j_t)})$.
2. Assume F_t is known.
   (a) For each α-particle, generate n_β number of particles of a second type, called $\beta^{(\alpha)}$-particles.
   (b) Weight each $\beta^{(\alpha)}$-particle according $g_2(\lambda_2, Q_t)$.
3. Maximize global likelihood $g_t^{z_t}$: obtain n_α particles by resampling the full set of $n_\alpha n_\beta$ particles according to the product of their weights
4. Update $R_t^{(j_t)}$ for each α-particle by applying h_2 to its $\beta^{(\alpha)}$ layer.
5. Marginalize over $\beta^{(\alpha)}$ layer yielding n_α number of α-particles with uniform weights.

Collectively, these steps form the branching mechanism for particles in the QSLAM filter. The word 'branching' refers to the concept of replacing the set of particles at the start of the t (parents) with a new set of particles (offspring) at the end of the iteration, after measurement data is received. The updates are performed such that the global likelihood $g_t^{z_t}$ is maximized.

The data association mechanisms h_1 and h_2 are written as a set of recursive equations which conform to both a Markov description of our state space and an iterative, numerical inference procedure. In the equations below, $h_1(\lambda_1, Z_t)$ updates the state variable $F_t^{(j)}$ given X_{t-1} (omitting t sub-script on j for ease of reading) and h_2 updates the state variable $R_t^{(j)}$ given F_t.

$$F_t^{(j)} := \begin{cases} F_{t-1}^{(j)}, & \tau_t^{(j)}, \varphi_t^{(j)} = 0 \\ h_1(\lambda_1, Z_t), & \text{otherwise} \end{cases} \tag{17}$$

$$R_t^{(j)} := h_2(F_t) \tag{18}$$

$$h_1(\lambda_1, Z_t) := \cos^{-1}(2P_t^{(j)}(\lambda_1, Z_t) - 1) \tag{19}$$

$$h_2(F_t) = \mathbb{E}[\{R_t^{(j),n}\}_{n=1}^{n_\beta} | F_t] \tag{20}$$

$$P_t^{(j)}(\lambda_1, Z_t) := \begin{cases} \left(1 - \frac{\lambda_1^{\tau_t^{(j)}}}{2}\right)\kappa_t^{(j)} + \left(\frac{\lambda_1^{\tau_t^{(j)}}}{2}\right)\gamma_t^{(j)}, \\ \text{for } \tau_t^{(j)}, \varphi_t^{(j)} \neq 0. \\ \kappa_t^{(j)}, \text{ for } \tau_t^{(j)} \neq 0, \varphi_t^{(j)} = 0. \\ \gamma_t^{(j)}, \text{ for } \tau_t^{(j)} = 0, \varphi_t^{(j)} \neq 0. \end{cases} \tag{21}$$

$$\kappa_t^{(j)} := \frac{\tau_{t-1}^{(j)}}{\tau_t^{(j)}}\kappa_{t-1}^{(j)} + \frac{1}{\tau_t^{(j)}} Y_t^{(j_t)} I_{(j_t=j)}, \quad \tau_t^{(j)} \neq 0, \kappa_0^{(j)} = 0 \tag{22}$$

$$\gamma_t^{(j)} := \frac{\varphi_{t-1}^{(j)}}{\varphi_t^{(j)}}\gamma_{t-1}^{(j)} + \frac{1}{\varphi_t^{(j)}} \hat{Y}_t^{(q_t)} I_{(q_t=j, q_t \in Q_t)}, \quad \varphi_t^{(j)} \neq 0, \gamma_0^{(j)} = 0 \tag{23}$$

$$\tau_t^{(j)} := \tau_{t-1}^{(j)} + I_{(j_t=j)}, \quad \tau_0^{(j)} = 0 \tag{24}$$

$$\varphi_t^{(j)} := \varphi_{t-1}^{(j)} + I_{(q_t=j, q_t \in Q_t)}, \quad \varphi_0^{(j)} = 0 \tag{25}$$

Here, I_X is the indicator function equally unity if X, and zero if not X. We see that $P_t^{(j)}$ is just an empirical Born probability estimate obtained by averaging the binary measurement data obtained at a single qubit. This data consists of averaging over $\tau_t^{(j)}$ physical measurements at qubit j, and $\varphi_t^{(j)}$ shared messages induced by

measuring the neighboring qubits of j. The role of shared messages is reduced as the number of physical measurements, $\tau_t^{(j)}$, increases, at a rate governed by the choice of λ_1. The data association mechanism h_2 locally updates the state variable $R_t^{(j)}$ as the expectation over resampled β-particles given the state at F_t. The dependence of h_2 on F_t is implicit, as β-particles are weighted via the $g_2(\lambda_2, Q_t)$ likelihood function for which it is assumed F_t is known.

With the above computations, QSLAM can be numerically implemented in software. Python-packages for the QSLAM algorithm, data generation scripts and simulation analysis are accessible via http://github.com/qcl-sydney/quantum_slam.