Active learning of quantum system Hamiltonians yields query advantage

Arkopal Dutt,[1,2,3,*] Edwin Pednault,[4] Chai Wah Wu,[5] Sarah Sheldon,[4] John Smolin,[4,3]
Lev Bishop,[4] and Isaac L. Chuang[2,3]

[1]*MIT-IBM Watson AI Lab, Cambridge, Massachusetts 02142, USA*
[2]*Center for Ultracold Atoms, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*
[3]*Department of Physics, Co-Design Center for Quantum Advantage,*
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA
[4]*IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, New York 10598, USA*
[5]*IBM Research AI, IBM T.J. Watson Research Center, Yorktown Heights, New York 10598, USA*

(Received 31 December 2021; revised 17 November 2022; accepted 2 May 2023; published 28 July 2023)

Hamiltonian learning is an important procedure in quantum system identification, calibration, and successful operation of quantum computers. Through queries to the quantum system, this procedure seeks to obtain the parameters of a given Hamiltonian model and description of noise sources. Standard techniques for Hamiltonian learning require careful design of queries and $O(\epsilon^{-2})$ queries in achieving learning error ϵ due to the standard quantum limit. With the goal of efficiently and accurately estimating the Hamiltonian parameters within learning error ϵ through minimal queries, we introduce an active learner that is given an initial set of training examples and the ability to interactively query the quantum system to generate new training data. To ensure applicability on near-term quantum hardware, the active learner operates in batch mode as opposed to sequentially, proposing batches of queries to be made during learning. We formally specify and experimentally assess the performance of this Hamiltonian active learning (HAL) algorithm for learning the six parameters of a two-qubit cross-resonance Hamiltonian on four different superconducting IBM quantum devices. Compared with standard techniques for the same problem and a specified learning error, HAL achieves more than a 95% reduction in queries required, and upwards of 33% reduction over a sequential active learner. Moreover, with access to prior information on a subset of Hamiltonian parameters and given the ability to select queries with linearly (or exponentially) longer system interaction times during learning, HAL can exceed the standard quantum limit and achieve Heisenberg (or super-Heisenberg) limited convergence rates during learning.

DOI: 10.1103/PhysRevResearch.5.033060

I. INTRODUCTION

Hamiltonian learning constitutes the problem of learning the Hamiltonian governing the dynamics of a quantum system given finite classical and quantum resources. This is a fundamental problem encountered in identification of quantum systems [1–3], operation of quantum information devices [4,5], validation of theoretical physical models, and has implications for computational bounds on quantum algorithms [6–8]. In the calibration of quantum computers alone, it is a significant step in each of the following tasks: system characterization, learning device parameters, different sources of noise, gate design [9], and control strategies for implementing robust quantum gates with high fidelity. Moreover, a quantum computer typically requires frequent recalibrations to account for drift in parameters over time requiring multiple iterations of some Hamiltonian learning routine.

The resource requirements for learning a generic many-body Hamiltonian rise exponentially with the system size [10]. Even for a fixed system size, however, the achievable learning error ϵ is fundamentally limited by the number of queries N made to the quantum system. In particular, through repeated system queries, N in general scales as ϵ^{-2} as a consequence of the central limit theorem. This is commonly referred to as the standard quantum limit (SQL) or shot noise limited scaling. Using quantum resources, however, a number of approaches have shown a much better *Heisenberg limited* scaling of $N \sim \epsilon^{-1}$. The Heisenberg limit is known to be fundamental [11–15], under a wide range of assumptions [16–19], but it is typically only saturated with the help of quantum resources.

This has been achieved using entanglement [20] such as NOON states [21,22], but can also be accomplished without entanglement under certain circumstances. For example in the problem of phase estimation [23], one has to estimate the phase ϕ in an unitary operator of form $U = \exp(-i\phi H)$ where H is a Hermitian operator and ϕ can also be interpreted as the strength of coupling in a Hamiltonian. It has been shown that Heisenberg limited scaling can be achieved

[*] arkopal@mit.edu

using multiround protocols using both adaptive measurements [24,25] and predetermined nonadaptive measurement sequences [26,27]. In contrast, there has not been such a detailed study in the case of learning a general many-body or a multiparametric Hamiltonian. This motivates understanding how might just classical resources be employed to solve the Hamiltonian learning problem with a scaling, which surpasses the SQL, and ideally achieves the Heisenberg limit.

Even if scalings higher than SQL cannot be achieved, it is still desirable to minimize resource requirements. This may be accomplished by changing the estimation procedure used for Hamiltonian learning in combination with engineered experiments. Fast Fourier transform (FFT) and linear regression are some of the traditional estimation methods, which still form the powerhouse of modern Hamiltonian learning strategies [28]. It has been shown that adopting alternate estimation methods such as Bayesian estimation [29], stochastic estimation [30], and neural-network based Hamiltonian reconstruction [4] can reduce resource requirements and improve scalability. We will call the reduction in resource requirements achieved by replacing one Hamiltonian learning strategy with another as *query advantage*. This will obviously depend on the two strategies being compared. We will call the strategy being replaced as the baseline. The baseline and our proposed replacements will be discussed in detail later. This brings us to the primary question we tackle in this paper: what is a common framework for Hamiltonian learning that can achieve query advantage even if scalings higher than SQL are not achieved, just using classical resources?

One effective framework for surpassing central limit theorem bounds when possible and/or achieving query advantage is *active learning*, e.g., using optimal experiment design. In [31], this idea was explored for quantum state tomography, process tomography, and Hamiltonian learning given a model, but in an offline manner. This has also been used to reduce experiment budget and propose different control schemes [32]. Active learning of a Hamiltonian is a more challenging problem than for quantum state tomography [33] where one optimizes over different measurements and process tomography [34] where one additionally optimizes over initial states, due to the additional control parameter of system time evolution. Active learning thus provides a general framework for making adaptive queries to the quantum system, comprised of initial state, measurements and system time evolution during Hamiltonian learning. In fact, [35] has shown that with adaptive feedback control, Heisenberg limited scaling can be reached in principle, but a recipe for this is only given for estimation of a single Hamiltonian parameter and the procedure requires prior information of the parameter. This was later extended to multiparameter Hamiltonians by [36]. A common ingredient of these works and the earlier mentioned multiround protocols for phase estimation is trading the cost of using physical quantum resources for cost in time resources [20].

A sequential active learner for Hamiltonian learning based on the criteria of Bayes risk was proposed as part of robust online Hamiltonian learning (ROHL) in [37] and later Qinfer [38]. However, this active learner operates in a sequential manner, proposing a query to be made one shot at a time. This limits its applicability to current hardware where batching queries (or quantum circuits) is essential to overcome costs of typically expensive computation of risk functions for different queries being considered, access over the cloud, compilation, system queues, and latencies between classical electronics and the quantum hardware. It is then necessary to ensure any active learner we introduce for Hamiltonian learning operates in batch mode.

In this paper, we introduce a Hamiltonian active learning algorithm (HAL) operating in batch mode based on the criteria of Fisher information, which is a way of measuring the information content of different queries and naturally appears in the bound on the errors achieved by a Hamiltonian learner. This resulting variant of HAL is called HAL-FI. We also introduce another variant of HAL for the task of predicting queries to the Hamiltonian, which uses the criteria of Fisher information ratio (FIR). We call the resulting active learning algorithm HAL-FIR. We demonstrate the performance of HAL-FI experimentally on IBM quantum devices, which are based on the superconducting cross-resonance (CR) gate. Compared with passive learning, which scales as SQL, we show that HAL-FI with a fixed space of queries also has an asymptotic scaling of SQL but is able to achieve a constant reduction of more than 96.9% in the number of queries required for a desired learning error in learning the two-qubit CR Hamiltonian on a 20-qubit IBM quantum device. HAL-FI can achieve more than 33% reduction in the number of queries required when compared to current standard methods used for Hamiltonian learning such as the sequential active learner of Qinfer. We finally show that queries involving exponentially growing system evolution time to the quantum devices suffices during learning to achieve Heisenberg limited scaling with HAL-FI when prior information is available. This is another example of trading physical quantum resources with time resources as highlighted before.

The paper is organized as follows. In Sec. II, we formally describe the problem of Hamiltonian learning, and the concept of an active learner. In Sec. III, we present the HAL algorithms of HAL-FI and HAL-FIR. To illustrate the performance of HAL-FI, we consider the example of calibrating CR gates on IBM quantum devices. In Sec. IV, we describe our experimental setup, provide a theoretical description of the Hamiltonian model of the CR gate and physical models of the different noise sources affecting the quantum devices. Further, we provide details of our experiments on evaluating the performance of HAL-FI. Finally in Sec. V, we compare the computational cost of different learners, describe the amount of query advantage that can be obtained using HAL-FI, and specify the conditions under which Heisenberg limited rate of convergence or even super-Heisenberg limited rate of convergence can be achieved. Specifically for CR gates, we show that HAL-FI can be used to learn an accurate Hamiltonian using only a fraction of the queries required by currently used methodologies, resulting in reduction of queries of around two or three orders of magnitude over currently used methods for particular learning tasks.

II. HAMILTONIAN LEARNING

In this section, we present a description of the problem of Hamiltonian learning for a general quantum system

(Sec. II A), and in the presence of different noise sources (Sec. II B). We will introduce the concept of an active learner in this context (Sec. II C). Notation used for this work will be defined as introduced.

A. Problem statement

This section of the paper describes the unknown Hamiltonian of interest, specifies our query setting, formally describes the different Hamiltonian learning tasks, and the estimators that are used for Hamiltonian learning.

1. Unknown Hamiltonian

Let H be a partially or fully unknown system Hamiltonian over n qubits. We can represent H in terms of the n-qubit Pauli operators as $H = \sum_{P\in\{\sigma_I,\sigma_X,\sigma_Y,\sigma_Z\}^{\otimes n}} c_P P$ with $\{\sigma_I, \sigma_X, \sigma_Y, \sigma_Z\}$ representing the single-qubit Pauli operators and where $c_P \in \mathbb{R}$ are the corresponding coefficients. We could also represent H as a Hermitian matrix in $\mathbb{C}^{2^n\times 2^n}$ under the usual computational basis.

We assume that we have a model for the unknown Hamiltonian denoted by $H(\boldsymbol{\theta})$, parametrized by a real vector $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^m$ of length m. We use Θ to denote the set of all possible values over the Hamiltonian parameters. The model may be derived from first principles based on the understanding of the physics of the quantum system or proposed through empirical observations. We further assume that the system Hamiltonian is time independent.

Let the unknown Hamiltonian of the quantum system be $H^\star$ and the true Hamiltonian parameters be $\boldsymbol{\theta}^\star$. We refer to the quantum system of whose Hamiltonian we wish to learn as an *oracle* that we can query and which returns measurement outcomes upon querying. We denote the random variable of query to the oracle by x and the resulting output by the oracle as the measurement outcome random variable y corresponding to a single shot of the qubits being readout in the standard computational basis. The pair of a query and its corresponding output is called an example and is denoted by (x, y). The alphabet of query x is referred to as the query space and we denote it by $\mathcal{Q}$. The distribution from which queries are sampled from $\mathcal{Q}$ is referred to as the *query distribution* and denoted by q. Commonly, $y \in \mathcal{Y} = \{0, 1\}^{n_r}$ where n_r is the number of qubits being readout. Our goal is to then learn the parameters $\boldsymbol{\theta}$ of the Hamiltonian with error ϵ from examples of the form $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$ while minimizing the number of queries made to the oracle or the query complexity N.

2. Specifying a query

We first describe what we mean by a query before specifying the different learning objectives. The query comprises three parts: measurement operators M, initial state preparation operators U, and control parameters t. A schematic of how the query is used in a quantum circuit to evolve a quantum system is shown in Fig. 1. In Fig. 2, we show the oracle of a quantum device receiving the input of queries of form $x = (M, U, t)$ and returning the corresponding outputs of a single shot of the qubits on the device being readout denoted by a binary string of length n_r: $y \in \{0, 1\}^{n_r}$.

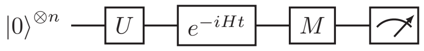

FIG. 1. Quantum circuit picture of how the query $x = (M, U, t)$ is used. The input to the quantum circuit is the zero state $|0\rangle^{\otimes n}$. Application of the preparation operator U on $|0\rangle^{\otimes n}$ is used to create an initial state before interacting with the system Hamiltonian H for time t. Finally, a measurement operator M is applied to the evolved state before measuring in the usual computational basis denoted by the meter. The output is a single shot of qubits being readout.

(a) Measurement operators. The measurement operators $M \in \mathcal{M}$ specify the admissible set of unitary operators that can be applied after evolving the system Hamiltonian and before measuring each qubit in the Z basis (as typical of current hardware). This in turn specifies the basis set $\{|\psi_m\rangle\}$ used to generate the final measurement observation results. The permissible set of M is typically constrained to be single-qubit operators or shallow circuits, because it is experimentally realistic to implement these on current hardware with high fidelity. For our applications, we will consider M to correspond to single-qubit operators corresponding to measuring under a Pauli basis.

(b) Initial state preparation operators. Together with the measurement operator, an initial state $|\psi_0\rangle$ must also be specified, in order to determine a measurement result from H. This initial state can in principle be any unit vector in the Hilbert space, but much like for measurement operators, we only allow a subset of possible states to be specified. We consider $|\psi_0\rangle = |0\rangle^{\otimes n}$ and allow for realistic single qubit unitary operators. A common initial state specification is to provide a unitary operator $U = \bigotimes_{i=1}^n U_i$ as a tensor product of single qubit operators such as single-qubit Pauli operators and Hadamard gates acting on each of the n qubits in the system. This then determines the U element of x. We denote the set of all considered preparation operators as $\mathcal{U}$

(c) Control parameters. The control parameters are the last element of x, and are typically a set of classical numbers. For example, one canonical control parameter is the time t for which H should be applied. Other control parameters may modulate interactions between qubits. In this work, we consider only the Hamiltonian evolution time t as a control parameter and denote the set of all possible evolution times as $\mathcal{T}$.

After defining the set of measurement operators $\mathcal{M}$, the set of preparation operators $\mathcal{U}$, and the set of Hamiltonian evolution times $\mathcal{T}$, the resulting query space is given by $\mathcal{Q} = \mathcal{M} \times \mathcal{U} \times \mathcal{T}$. We require $\mathcal{Q}$ to be *complete*, i.e., there exist queries in $\mathcal{Q}$ that are informative about each of the Hamiltonian parameters $\boldsymbol{\theta}_i$ so that Hamiltonian learning can succeed. As we will see later in our discussion on active learning in Sec. II C 2, completeness of $\mathcal{Q}$ is equivalent to the condition of there existing a set of queries or query distribution for which the resulting Fisher information matrix is full rank and invertible.

3. Learning framework

Having fully described a query, we are now in a position to formalize the problem of Hamiltonian learning. While doing

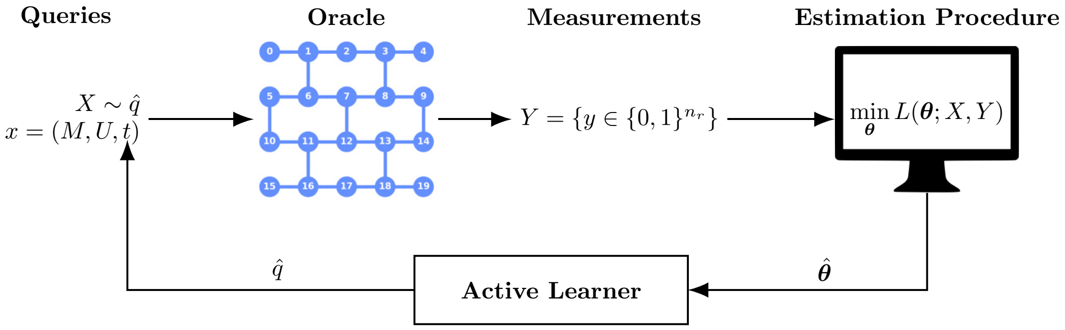

FIG. 2. Schematic of Hamiltonian learning with an active learner: The *oracle* constitutes the unknown Hamiltonian and noise sources, with the true parameter vector of $\boldsymbol{\theta}^\star$, which is unknown and needs to be learned. Here, we show the device coupling map of the 20-qubit IBM quantum device *ibmq_boeblingen*. Learning is carried out on training examples of the form (x, y) where x are queries inputted to the oracle specifying the measurement observable M, preparation operator U and system time evolution t, and y are the corresponding measurement outcomes outputted by the oracle. An illustrative quantum circuit picture of the oracle in the noiseless case is shown in Fig. 1. The set of queries inputted to the oracle is denoted as X and the corresponding set of measurement outcomes outputted by the oracle as Y. An estimation procedure is run on the training examples of (X, Y) to learn a model parameter estimate $\hat{\boldsymbol{\theta}}$. The complete top row corresponds to how a passive learner operates and is also called open-loop Hamiltonian learning. We add a feedback loop to introduce an active learner, which uses the current estimate of model parameters $\hat{\boldsymbol{\theta}}$ to prescribe the distribution $\hat{q}$ from which queries to the oracle should be sampled from next. This process is then repeated during learning until an accurate estimate $\hat{\boldsymbol{\theta}}$ is obtained. The Hamiltonian active learning (HAL) algorithm introduced in this paper comprises the estimation procedure and active learner shown in this schematic. The active learner is described in Sec. II C and the HAL algorithm in Sec. III.

so, we draw parallels to and introduce language from machine learning and statistical learning theory.

We are given a query space $\mathcal{Q} = \mathcal{M} \times \mathcal{U} \times \mathcal{T}$ constructed as discussed above and a space over the measurement outcomes $\mathcal{Y} = \{0, 1\}^{n_r}$. We consider the class of Hamiltonian models $\mathcal{H}$. A Hamiltonian $H \in \mathcal{H}$ defines a map from $\mathcal{Q}$ to the set of measurement outcomes taking values in $\mathcal{Y}$ by Born's rule. In particular we assume the availability of a model description $\mathcal{H} = \{H(\boldsymbol{\theta}) | \boldsymbol{\theta} \in \boldsymbol{\Theta}\} \subset \mathbb{R}^m$ parametrized by the vector $\boldsymbol{\theta}$ and where Θ is considered to be the space over the parameters. When no prior information is available to constrain the parameter space, we can consider $\Theta = \mathbb{R}^m$. We denote the true Hamiltonian as $H^\star$ and assume that $H^\star \in \mathcal{H}$, i.e., it is realizable. We denote the parameter vector of the true Hamiltonian as $\boldsymbol{\theta}^\star$. Our goal is to learn an estimate of $\boldsymbol{\theta}^\star$, which we denote by $\hat{\boldsymbol{\theta}}$. With regards to notation, we use $\boldsymbol{\theta}$ whenever we make statements that are true for any parameter in Θ.

For the parameter vector $\boldsymbol{\theta}$, the label (or measurement outcome) y given query $x = (M, U, t)$ is produced with the conditional probability

$$p_{\mathrm{y|x}}(y|x; \boldsymbol{\theta}) = \sum_z |\langle yz| M e^{-iH(\boldsymbol{\theta})t} U |0^{\otimes n}\rangle|^2 \quad (1)$$

as per Born's rule and assuming the absence of noise. The summation is over the hidden measurement outcomes of the $(n - n_r)$ qubits that are not read out, which we denote by z and $|\cdot|$ is used to denote the absolute value.

Given the dataset of N training examples $D = \{(x^{(i)}, y^{(i)})\}_{i \in [N]}$, the goal is to learn the Hamiltonian parameter vector $\boldsymbol{\theta}$, which characterizes the conditional probability [Eq. (1)] over measurement outcomes in $\mathcal{Y}$ given queries in $\mathcal{Q}$. This is a supervised learning task for which a suitable loss function is

$$\mathrm{Loss}(\boldsymbol{\theta}; D) = \mathcal{L}(D; \boldsymbol{\theta}) + R(\boldsymbol{\theta}) \quad (2)$$

where $\mathcal{L}$ is a suitable error function for assessing the chosen model on the training dataset, and R is a regularization function penalizing model complexity. The latter is chosen to incorporate prior information, enforce conditions such as sparsity through ℓ_1 norm and generalize to unseen data. The parameter estimate is determined by minimizing the loss function $\hat{\boldsymbol{\theta}} = \mathrm{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} L(\boldsymbol{\theta}; D)$. After obtaining an estimate $\hat{\boldsymbol{\theta}}$, it is common to evaluate the performance against a testing dataset, which includes examples not seen during training. This testing dataset, which we denote as $D_{\mathrm{test}} = \{(x^{(i)}, y^{(i)})\}_{i \in [N_{\mathrm{test}}]}$ contains N_{test} queries and their corresponding outputs. The N_{test} queries are sampled from the query space $\mathcal{Q}$ according to a testing distribution, which we denote by p_{test}. This allows us to further distinguish between the problems of *model inference* and *prediction* against p_{test}.

(a) Model inference. The goal is to learn a parameter estimate $\hat{\boldsymbol{\theta}}$ within a prescribed learning error ϵ with the fewest queries N, without making special considerations for the testing distribution p_{test}, which may change or may be unknown. We use root mean squared error (RMSE) as $\mathcal{L}$ in Eq. (2),

$$\mathrm{RMSE}(\hat{\boldsymbol{\theta}}; \boldsymbol{\xi}) = \left(\sum_{i=1}^{m} \mathbb{E}\left[\left(\frac{\hat{\theta}_i}{\xi_i} - \frac{\theta_i^\star}{\xi_i} \right)^2 \right] \right)^{1/2} \quad (3)$$

where $\boldsymbol{\theta}^\star$ is the parameter vector corresponding to the unknown Hamiltonian $H^\star$, and $\boldsymbol{\xi}$ is the vector to nondimensionalize and normalize $\boldsymbol{\theta}$. This is done to account for different relative magnitudes of each parameter component $\theta_i^\star$ and to ensure that the RMSE does not explode due to contributions of few parameter components. These normalization factors $\boldsymbol{\xi}$ can be obtained during estimation or be available through prior information. When an unbiased estimator is used, $\mathcal{L}$ may be reduced to being the square root of the variance, $[\sum_j \mathrm{Var}(\theta_j/\xi_j)]^{1/2}$. It is typically hard to ensure that an estimator is unbiased, even though this is desired. To find $\hat{\boldsymbol{\theta}}$ that minimizes the RMSE given D, we use estimation methods based on the asymptotically unbiased maximum likelihood estimator (MLE) and Bayesian estimation, all of which we will discuss in Sec. II A 4.

(b) Prediction against a testing distribution p_{test}. The goal is to learn an estimate $\hat{\boldsymbol{\theta}}$ that will allow us to perform well on predicting the likelihood function of measurement outcomes given queries sampled from p_{test}. In such a scenario, $\mathcal{L}$ is still the RMSE but the performance of the estimate is assessed through the testing error,

$$\text{Testing Error} = \mathbb{E}_{x\sim p_{\text{test}}, y\sim p_{\mathrm{y|x}}(\cdot|x;\boldsymbol{\theta}^\star)}[L(y|x;\hat{\boldsymbol{\theta}}) - L(y|x;\boldsymbol{\theta}^\star)], \tag{4}$$

where we have denoted the negative log-likelihood $-\log p_{\mathrm{y|x}}(y|x;\boldsymbol{\theta})$ in shorthand by $L(y|x;\boldsymbol{\theta})$ for a general measurement outcome y and query x given model parameters $\boldsymbol{\theta}$. We have chosen the testing error as the expectation of the difference in negative log-likelihood when using the estimate $\boldsymbol{\theta}$ and truth $\boldsymbol{\theta}^\star$ with respect to the testing distribution p_{test}. This difference in negative log-likelihood is also often called the log-likelihood ratio [39]. It is then desired to learn $\hat{\boldsymbol{\theta}}$ such that the testing error is lower than a given error ϵ. The error ϵ here may need to be specified relative to how the negative log-likelihood scales, which will depend on the Hamiltonian of interest and the queries themselves.

Given N training examples, we then say that we have succeeded at model inference (prediction) if we are able to produce an estimate $\hat{\boldsymbol{\theta}}$ such that the RMSE (testing error) is bounded by some error parameter ϵ. We want to accomplish these learning tasks by minimizing the resource requirements or number of queries N.

We distinguish between the two problems of model inference and prediction against p_{test} as the ideal training datasets for these two learning tasks may come from two different distributions. This will influence how the training data is chosen when we introduce an active learner into Hamiltonian learning, as we will discuss in Sec. III. The performance of the Hamiltonian learning algorithms in tackling the learning Problem II A 3 b will also be useful for cross-validation and testing the robustness of these methods.

4. Estimators

We now discuss the different types of estimators that may be adopted for learning an estimate $\boldsymbol{\theta}$ of the Hamiltonian parameters from the N measurement outcomes $Y = \{y^{(i)}\}_{i\in[N]}$ obtained from the quantum system upon N queries $X = \{x^{(i)}\}_{i\in[N]}$. Collectively, we refer to them as the training data $D = (X, Y) = \{(x^{(i)}, y^{(i)})\}_{i\in[N]}$. The active learner for Hamiltonian learning we introduce in this paper will be designed for use with the following maximum likelihood estimator (MLE) but is also compatible with the other estimators mentioned here.

(a) Maximum likelihood estimation. The maximum likelihood estimator (MLE) determines an estimate of the parameters $\hat{\boldsymbol{\theta}}$ from the data D through

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} -\log p_{\mathrm{y|x}}\big(y^{(i)}|x^{(i)};\boldsymbol{\theta}\big), \tag{5}$$

where $-\log p_{\mathrm{y|x}}(y^{(i)}|x^{(i)};\boldsymbol{\theta})$ is the negative log-likelihood of the measurement outcome $y^{(i)}$ given the query $x^{(i)}$ and when considering the model parameters $\boldsymbol{\theta}$. The optimization problem of Eq. (5) is typically nonconvex and thus specific procedures may be required for avoiding local minima and converging to the global minimum.

(b) Regression. Estimation based on regression is used as a proxy for the MLE or sometimes even as part of the first step in a procedure to solve the optimization of MLE [Eq. (5)]. Let the empirical likelihood computed from measurement outcomes corresponding to query x be given by $\hat{p}_{y|x}$. Suppose we have a model $l(x;\boldsymbol{\theta})$ (e.g., Gaussian or sinusoidal) of the likelihood function for query x given parameters $\boldsymbol{\theta}$. We can then obtain an estimate by performing least squares,

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, ||\hat{p}_{Y|X} - l(X;\boldsymbol{\theta})||_2^2, \tag{6}$$

where we have denoted $\hat{p}_{Y|X}$ as the empirical likelihood computed for different queries in queries X using corresponding measurement outcomes in Y from D. Note that Eq. (6) could be further improved by weighting the empirical likelihoods by their variances due to shot noise.

(c) Bayesian estimation. Instead of the frequentist approach above, we can also adopt a Bayesian approach during estimation. The Hamiltonian parameters $\boldsymbol{\theta}$ are treated as a random vector whose distribution is updated according to Baye's rule,

$$p(\boldsymbol{\theta}|D) = \frac{p(Y|\boldsymbol{\theta}, X)p(\boldsymbol{\theta})}{p(Y|X)}, \tag{7}$$

where $p(Y|\boldsymbol{\theta}, X)$ is the likelihood function over measurement outcomes Y given queries X and parameter vector $\boldsymbol{\theta}$, $p(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$, and $p(\boldsymbol{\theta}|D)$ is the posterior distribution after incorporation of training data $D = (X, Y)$. Often, we will use the Baye's rule iteratively and in that case the posterior from one step becomes the prior for the next step. When a single point parameter estimate is desired, we will use the posterior mean $\hat{\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{\theta}|D}[\boldsymbol{\theta}]$, which is optimal for minimizing RMSE [27].

Computing the denominator in Eq. (7) and the mean over the posterior distribution is computationally expensive and usually intractable in practice. This computational burden is reduced if structural information about the distributions in Eq. (7) are known (e.g., Gaussian) or can be reliably enforced. This is done in Bayesian estimation methods such as the Kalman filter [40] and Gaussian mixture model based filter [41]. For our applications, we will consider sequential Monte Carlo methods (also called particle filtering) [42].

We note that Eq. (5) is an optimization problem that may be solved in several ways, and similarly there may be different strategies to evaluate Eq. (7) in Bayesian estimation. The different programs or algorithms to solve the optimization problem (or Baye's rule) are commonly called *optimization procedures* or *estimation procedures*. Different procedures have different properties such as rate of convergence and optimality such as guarantee of a local or global minimum. In order to improve convergence or search direction, it is common for procedures to require first or second gradient information. The estimation procedure will need to be specified along with the formulated optimization problem at hand for a complete specification of the learning algorithm or learner.

B. Learning in the presence of noise

Previously, in Sec. II A, we gave formal statements of the Hamiltonian learning problem in the absence of noise assuming an ideal oracle. However, the oracle is usually noisy due to the presence of different noise sources affecting the quantum system. In this section, we describe the problem of learning in the presence of these noise sources.

Common noise sources include readout noise, decoherence, and imperfect control of the quantum system. State preparation and measurement (SPAM) errors are already accounted for by considering these noise sources. Classical SPAM errors encountered are included in the readout noise and the errors in implementing state preparation or measurement operators fall under imperfect control.

(a) Readout noise. The readout line of the qubit measurement outcomes y is also a classical communication channel and hence suffers from bit-flip errors. The readout noise can then be modeled as a classical bit flip channel where the true measurement outcome $y_i \in \{0, 1\}$ of the ith qubit may be flipped. We denote the observed measurement outcomes' random variable by $\tilde{\mathrm{y}}$, which can be interpreted as a noisy observation of y. We denote the conditional probability of observing $\tilde{y}$ given measurement outcome y, which is hidden from us as $p_{\tilde{\mathrm{y}}|\mathrm{y}}(\tilde{y}|y)$. Note that in general we expect the readout channel to be asymmetric, i.e., $p_{\tilde{\mathrm{y}}|\mathrm{y}}(1|0) \neq p_{\tilde{\mathrm{y}}|\mathrm{y}}(0|1)$, and this is accounted for in the readout noise model. The probability of observing a noisy measurement outcome $\tilde{y}_i$ of the ith qubit given query x is then given by

$$p_{\tilde{\mathrm{y}}_i|\mathrm{x}}(\tilde{y}_i|x;\boldsymbol{\theta}) = p_{\tilde{\mathrm{y}}_i|\mathrm{y}}(\tilde{y}_i|\tilde{y}_i)p_{\mathrm{y}_i|\mathrm{x}}(\tilde{y}_i|x;\boldsymbol{\theta}) + p_{\tilde{\mathrm{y}}_i|\mathrm{y}}(\tilde{y}_i|1-\tilde{y}_i) \times p_{\mathrm{y}_i|\mathrm{x}}(1-\tilde{y}_i|x;\boldsymbol{\theta}) \tag{8}$$

where $p_{\mathrm{y}|\mathrm{x}}(y|x;\boldsymbol{\theta})$ is given by Eq. (1).

(b) Imperfect control strategies. In general, a Hamiltonian can be decomposed as

$$H(t) = H_d + \sum_{k=1}^{K} H_k(t, u_k(t)) \tag{9}$$

where H_d is the drift/free part of the Hamiltonian that is internal to the system and cannot be controlled externally. The terms H_k corresponds to the parts of the Hamiltonian that can be controlled using the control function $u_k(t)$. Depending on the quantum architecture, the control function can be realized as microwave pulses for superconducting qubits, optical pulses for ion traps, etc.

Target operators U_f are then obtained starting from the identity matrix I by implementing controls $u_k(t)$ over time duration $[0, T]$ as

$$U_f = \mathbb{T}\left[\exp\left(-i\int_0^T \left[H_d + \sum_{k=1}^{K} H_k(t, u_k(t))\right]dt\right)\right] \tag{10}$$

where $\mathbb{T}$ is the time-ordering operator. These target operators U_f can be preparation operators, measurement operators, or different gates, e.g., the Clifford gates. Imperfections in these pulses will lead to errors in U_f. For example, the strength of the qubit-qubit interactions is sensitive to variations in the pulse amplitudes. Strong driving can lead to leakage of states outside of the computational subspace. Moreover, bandwidth effects or dispersion can cause leading or trailing edge distortions in the pulse shapes that can lead to errors in the unitary operator implemented.

(c) Decoherence. Let us consider the unitary operator of $U(t) = e^{-iHt}$. The application of this unitary operator is accompanied by decoherence on a real quantum system due to interactions with its environment. We model this as a depolarizing channel $\mathcal{E}$,

$$\mathcal{E}(\rho(t)) = (1 - p_d(t))\rho(t) + p_d(t)\frac{I}{2^n} \tag{11}$$

where $\rho(t) = U(t)\rho(0)U(t)^{\dagger}$ is the state obtained on application of the unitary $U(t)$ to the initial state $\rho(0)$, $p_d(t)$ is the probability of the state being depolarized, and $I/2^n$ is the maximally mixed state. We have assumed the case of complete depolarization here. To obtain a functional form of $p_d(t)$, we note that up to first order, depolarization events can be assumed to arrive under a Poisson process with rate μ. By a depolarization event, we refer to the occurrence of an error that completely randomizes the quantum state. As the time between Poisson events follows an exponential distribution, we can write

$$p_d(t) = 1 - \exp\left(-\frac{t - t_0}{\mu}\right) \tag{12}$$

where t_0 denotes the starting time of the experiment. Denoting the probability of measurement outcomes y given a query x under depolarization errors as $p_{\mathcal{E}}(y|x;\boldsymbol{\theta})$, we have

$$p_{\mathcal{E}}(y|x;\boldsymbol{\theta}) = \exp\left(-\frac{t-t_0}{\mu}\right)p_{\mathrm{y}|\mathrm{x}}(y|x;\boldsymbol{\theta}) + \frac{1}{2^n}\left(1 - \exp\left(-\frac{t-t_0}{\mu}\right)\right) \tag{13}$$

where $p_{\mathrm{y}|\mathrm{x}}(y|x;\boldsymbol{\theta})$ is the probability of measurement outcome y assuming no depolarization given by Eq. (1). The rate μ can be related to the amplitude relaxation time T_1 and dephasing time T_2 of the qubits on the quantum system. We will describe this in Sec. IV C for the IBM quantum devices considered for application of Hamiltonian learning.

Let the collective set of parameters associated with the different noise models so far described be denoted by $\boldsymbol{\zeta}$. The measurement outcomes y from the oracle will then be a function of the queries x and the true parameters $(\boldsymbol{\theta}^{\star}, \boldsymbol{\zeta}^{\star})$. In order to obtain estimates for $\hat{\zeta}$ in addition to $\hat{\boldsymbol{\theta}}$, we solve the following modified optimization problem over the training examples

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\zeta}}) = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}, \boldsymbol{\zeta}\in Z}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\big(x^{(i)}, y^{(i)}; \boldsymbol{\theta}, \boldsymbol{\zeta}\big) + R(\boldsymbol{\theta}, \boldsymbol{\zeta}). \tag{14}$$

However, this increases the computational cost of the estimation procedure due to the increase in the number of parameters and the corresponding search space. In practice, prior calibration data can be used to obtain estimates of $\hat{\boldsymbol{\zeta}}$, which are then used in Eq. (2) to obtain $\hat{\boldsymbol{\theta}}$.

Hence, the Hamiltonian learning Problems II A 3 a and II A 3 b can be restated assuming additional information of the noise model parameters $\hat{\xi}$ when we have access to a noisy oracle. When describing our application of Hamiltonian learning, we will describe both the Hamiltonian of interest in Sec. IV A and the specific noise sources in Sec. IV C.

C. Active learning of Hamiltonians

This subsection introduces the concept of active learning and how an active learner can be used in the context of Hamiltonian learning. We begin by giving a quick overview of different types of learners, and describe how active learning differs from passive learning and online learning. This is followed by an overview of the different active learning (AL) strategies that have been proposed in the literature. We describe which AL strategy seems best suited for Hamiltonian learning and particularly when using the maximum likelihood estimator (MLE) (Sec. II A 4 a). AL strategies based on Fisher information (FI) for Problem II A 3 a and Fisher information ratio (FIR) for Problem II A 3 b seem to be notably appropriate. In doing so, we establish criteria for evaluating the performance of an AL algorithm, and observe that AL algorithms are known to solve some learning problems faster than passive learners. Moreover, under some circumstances, AL algorithms can exceed the central limit theorem bounds.

1. Types of learners

An overview of different learners often considered for learning tasks is given below. We will use the passive learner as a baseline and the active learner is the focus of our paper. A description of an *offline learner* and *online learner* is given for completeness and to distinguish an active learner from them.

(a) Offline learner. In offline learning, all of the training data is given to the learner at once and a model is learned. The training dataset may be made available to the learner or collected by sampling queries from an arbitrary distribution and then obtaining outputs from an oracle using these as inputs. This is the learning paradigm under which most machine learning tasks operate in.

(b) Online learner. In online learning, the training data is made available in a sequence, typically one at a time by a referee. A query x_t is made available to the learner at the tth round in a sequence after which the learner constructs an estimate of the output $\hat{y}_t$ to this query. The learner provides $\hat{y}_t$ to the referee, and suffers a loss that depends on $\hat{y}_t$ and the actual output. The learner is then provided with feedback by the referee, which the learner can then use to update the model. In this case, the queries that are provided to the learner by the referee may be adversarial or adaptive to the learner's behavior. The learner has no control over the query distribution from which these queries arrive.

(c) Passive learner. In the learning problems described in Sec. II A, we did not specify the distribution from which the queries x are sampled from. In open-loop Hamiltonian learning, the query distribution remains fixed during learning and all the queries to the oracle are sampled from this distribution. When no prior information is available, it is common to set the query distribution to the uniform distribution over the query space $\mathcal{Q}$. We will refer to this setting as *passive learning* through out this paper. Combined with a specification of an estimation procedure, the passive learner will serve as a baseline to the active learner, which we introduce in the next section.

(d) Active learner. In active learning (AL), the learner has access to the query space $\mathcal{Q}$ and the ability to select queries or decide the query distribution during training using the current estimates of the model parameters $\hat{\boldsymbol{\theta}}$. This is accomplished by introducing a feedback into the open loop Hamiltonian learning approach shown earlier in Fig. 2. Based on the current estimate $\hat{\boldsymbol{\theta}}$ and the queries made so far combined with their respective outcomes, the active learner proposes a query distribution from which queries should be selected from to send to the oracle. These queries may be sent to the oracle in a sequential manner one at a time or in batches. In Sec. II C 2, we discuss different criteria used for query selection or proposing query distributions.

Here, we distinguish a passive learner from an offline learner only in the number of rounds of training data collection. An offline learner is given access to a complete training dataset or is allowed to collect it through queries to an oracle in one round. On the other hand, a passive learner has continued access to the oracle and is allowed to collect training data until the learning task has been accomplished. The primary difference between the active learner and the online learner is that the former has control over the query distribution from which it will select queries to input to the oracle.

2. Query criteria for active learner

There have been multiple criteria proposed for query selection but are usually subdivided into the two categories of informativeness and representativeness. Criteria based on informativeness aim to select queries that will reduce the uncertainty of the statistical model and include uncertainty sampling [43,44], query-by-committee [45,46], and margin [47]. On the other hand, the goal of representativeness [48,49] is to ensure selection of queries that exploit the structure of the underlying distribution and are diverse. There has also been exploration into combining the criteria of informativeness and representativeness [50,51].

Multiple query criteria used in active learning in practice are based on heuristics and empirical evidence [52]. Here, we choose the informativeness criteria of Fisher information (FI) and Fisher information ratio (FIR) as they have direct relationships with the different learning problems we introduced in Sec. II A. This allows us to provide guarantees on the performance of our AL strategy. Later, we discuss how we can ensure that representative queries are also selected.

We introduce some notation before discussing the query criteria for our AL strategy. Let us denote the Fisher information matrix of a particular query $x \in \mathcal{Q}$ as $\mathcal{I}_x(\boldsymbol{\theta})$ where the (i, j)th element of the matrix is given by

$$\mathcal{I}_x(\boldsymbol{\theta})[i, j] = \mathbb{E}\left[\frac{\partial \log p_{\mathrm{y}|\mathrm{x}}(y|x;\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial \log p_{\mathrm{y}|\mathrm{x}}(y|x;\boldsymbol{\theta})}{\partial\theta_j}\right] \quad (15)$$

and where the expectation is taken with respect to $p(y|x;\boldsymbol{\theta})$. The Fisher information matrix is equivalently written as $\mathcal{I}_x(\boldsymbol{\theta}) = \mathbb{E}[SS^T]$ where $S = \partial \log p_{\mathrm{y}|\mathrm{x}}(y|x;\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ is commonly called the score vector. Instead of selecting one query at a time, we often require the active learner to select a distribution over the query space $\mathcal{Q}$, which we will call the query distribution. The Fisher information matrix corresponding to q will then be given by

$$\mathcal{I}_q(\boldsymbol{\theta}) = \mathbb{E}_q[\mathcal{I}_x(\boldsymbol{\theta})] = \sum_{x\in\mathcal{Q}} q(x)\mathcal{I}_x(\boldsymbol{\theta}) \tag{16}$$

where the summation can be replaced by an integral in the case of a continuous query space. If the parameters describing the Hamiltonian model are $\boldsymbol{\theta} \in \mathbb{R}^m$, then $\mathcal{I}_q(\boldsymbol{\theta}) \in \mathbb{R}^{m\times m}$. Let us now describe the different query criteria in the context of the different learning tasks.

(a) RMSE of parameters. If the learning objective is to learn the parameters with small RMSE (Problem II A 3 a), a natural query optimization strategy is obtained by noting the Cramer-Rao bound for unbiased estimators [53],

$$\text{Cov}(\boldsymbol{\theta}) \geqslant \frac{1}{N}\mathcal{I}_q^{-1}(\boldsymbol{\theta}), \tag{17}$$

$$\sum_i \text{Var}(\theta_i) \geqslant \frac{1}{N}\text{Tr}\big(\mathcal{I}_q^{-1}(\boldsymbol{\theta})\big). \tag{18}$$

Combining the Cramer-Rao bound with the fact that the parameter estimates converges in probability $\hat{\boldsymbol{\theta}} \to \boldsymbol{\theta}^\star$ for an unbiased estimator such as MLE and the distribution converges in law as $\sqrt{N}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star) \to \mathcal{N}(0, \mathcal{I}_q^{-1}(\boldsymbol{\theta}^\star))$ [54,55], we get that MLE is an asymptotically efficient estimator with the efficiency equal to the Fisher information over the training examples [39]. An optimal query distribution can then be obtained through the following query optimization:

$$q^\star = \arg\min_{q\in\mathcal{P}} \text{Tr}\big(\mathcal{I}_q^{-1}(\boldsymbol{\theta}^\star)\big) \tag{19}$$

where $\mathcal{P}$ is the family of all valid probability distributions over the query space $\mathcal{Q}$. We also note that for $q^\star$, $\mathcal{I}_{q^\star}$ must necessarily be invertible and hence full rank. This ensures that the queries inform us about all the Hamiltonian parameters of interest. Also for an unbiased and consistent estimator, the Cramer-Rao bound is likely to be saturated in the limit of large number of queries.

(b) Testing accuracy against testing distribution. When the learning objective is to minimize the expected log-likelihood error against a testing distribution p_{test} (learning Problem II A 3 b), we use an active learning strategy based on Fisher Information ratio (FIR) $\text{Tr}(\mathcal{I}_q^{-1}(\boldsymbol{\theta})\mathcal{I}_{p_{\text{test}}}(\boldsymbol{\theta}))$. The name FIR comes from the scalar case where it can be viewed a ratio of the Fisher information corresponding to the query distribution and that corresponding to the testing distribution. The use of FIR for this learning task can be motivated by noting the following inequality [39]:

$$\mathbb{E}_{x\sim p_{\text{test}}, y\sim p_{y|x}(\cdot|x;\boldsymbol{\theta})}[\text{Var}_{\text{D}\sim q(x)p_{y|x}(y|x;\boldsymbol{\theta})}[\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{D}}; x, y) - \mathcal{L}(\boldsymbol{\theta}; x, y)]]$$
$$\leqslant \frac{1}{N}\text{Tr}\big(\mathcal{I}_q^{-1}(\theta)\mathcal{I}_{p_{\text{test}}}(\boldsymbol{\theta})\big), \tag{20}$$

where the left-hand side is the expected variance of the asymptotic distribution of the log-likelihood ratio, which can be viewed as a testing error and the right-hand side involves the FIR. Minimizing the upper bound would then allow us to control the testing error and hence this suggests using the following query distribution:

$$q^\star = \arg\min_{q\in\mathcal{P}} \text{Tr}\big(\mathcal{I}_q^{-1}(\boldsymbol{\theta}^\star)\mathcal{I}_{p_{\text{test}}}(\theta^\star)\big). \tag{21}$$

We note the Fisher information ratio is related to the Fisher information matrices of the query distribution $\mathcal{I}_q$ and testing distribution $\mathcal{I}_{p_{\text{test}}}$ through the following inequality:

$$\text{Tr}\big(\mathcal{I}_q^{-1}(\boldsymbol{\theta})\mathcal{I}_p(\boldsymbol{\theta})\big) \leqslant \text{Tr}\big(\mathcal{I}_q^{-1}(\boldsymbol{\theta})\big)\cdot\text{Tr}(\mathcal{I}_{p_{\text{test}}}(\boldsymbol{\theta})). \tag{22}$$

Thus, we recover the query optimization of Eq. (19) when the testing distribution is unknown.

In quantum tomography, such query criteria have been applied in optimal experiment design (OED) or adaptive quantum tomography. Fisher information has been used as a query criteria for offline OED in quantum state tomography [33]. Even earlier, an active learner based on Shannon entropy also known as maximum uncertainty sampling was considered in [56]. An AL strategy based on Shannon information combined with Bayesian estimation was proposed in [57] for the selection of measurement operators during quantum state tomography. Fisher information was again used in [34] where OEDs were analyzed for a family of qubit channels over different design problems. In Hamiltonian learning, Fisher information has been used to comment on heuristic strategies for OED [58] and has been combined with Bayesian estimation to produce a sequential active learner [37].

3. *Active learning strategy*

Implicit in the above descriptions of the query criteria is the idea of proposing a query distribution rather than selecting one query at a time during active learning. This demarcates sequential active learning where one query is chosen at a time from batch mode active learning where a batch of queries sampled from a query distribution are selected to be inputted to the oracle. Moreover, combining FI/FIR query criteria with batch mode active learning [59] ensures that representative queries are chosen as well.

In what follows, we describe the batch-mode active learning scheme that forms the basis of the AL algorithm for Hamiltonian learning that we discuss in Sec. III. Given a budget of N queries, the training is divided into multiple rounds. We index each round of the training process as i and denote the batch size as N_b. The number of queries made till the ith round (inclusive) is denoted as $N_{\text{tot}}^{(i)}$. In each round, a batch of queries is sampled from the optimal query distribution based on the current parameters' estimate $\hat{\boldsymbol{\theta}}$ and then this estimate is updated using the measurement outcomes of the queries. This is then repeated until all of the budget has been expended. We denote the estimate in the ith round by $\hat{\boldsymbol{\theta}}^{(i)}$ and the optimal query distribution in the ith round based on $\hat{\boldsymbol{\theta}}^{(i)}$ by $q^{(i)}$. The query distribution at the very beginning of the training process $q^{(0)}$ is determined from any available prior information of the parameters or else set to be the uniform distribution over the query space.

What should be the size of the initial set of queries $N_{\text{tot}}^{(0)}$? Some suggestions are given in [60] based on a finite-sample analysis for logistic regression but these do not suffice for the application considered in this paper. Qualitatively, one hopes that $N_{\text{tot}}^{(0)}$ is high enough such that the parameter estimate $\hat{\boldsymbol{\theta}}^{(0)}$ lies close to the true parameter value $\boldsymbol{\theta}^\star$ and in a convex basin of the asymptotic negative log-likelihood loss function. However, setting $N_{\text{tot}}^{(0)}$ to a very high value may not allow us take advantage of the presence of an active learner and the savings it can provide.

Additionally, it may be advantageous to adaptively change the query space for exploration from one batch to the next. It is not necessary for the query space to remain static or unchanged [52] during active learning. In fact, there is an element of adaptively changing the *search* space in many prominent algorithms. In sparse fast Fourier transform [61,62], the bins of frequencies are randomly chosen with each iteration in the algorithm. In the related machine learning tool of reinforcement learning, action spaces are changed to eliminate actions [63], and to generalize over time by parametrizing them [64] or embedding them in a continuous action space [65].

Adaptively changing the query space is particularly compelling for the application of active learning to the Hamiltonian learning problem, because evidence suggests that it may result in so-called *Heisenberg*-limited scaling of the number of queries as we noted in Sec. I. We will also make a case for why we expect the active learner, which we will introduce for Hamiltonian learning in Sec. III to achieve Heisenberg-limited scaling when possible.

The computational cost of the batch-mode AL scheme is determined by the number of rounds of batches issued and the computational cost of solving the query optimization problems of Eqs. (19) and (21). Solving these directly can be challenging but fortunately the query optimizations can be reformulated as semidefinite programs (SDP) [39] under the assumptions of differentiability of the log-likelihood function, and invertibility of the Fisher information matrix for query distributions in a compact space around the optimal query distribution and around the uniform distribution. When using the query criteria of Fisher information ratio (FIR) [Eq. (21)], the optimization problem [39,60] is

$$\arg\min_{\alpha_1,\ldots,\alpha_d} \sum_{i=1}^{m} \alpha_i \quad \text{such that} \sum_{x\in\mathcal{Q}} q(x) = 1, \text{ and}$$
$$\begin{bmatrix} \mathcal{I}_q(\boldsymbol{\theta}) & \mathbf{e}_j \\ \mathbf{e}_j^T & \alpha_j \end{bmatrix} \succcurlyeq 0,\ j \in [m] \tag{23}$$

where we have introduced m auxiliary variables $\alpha_1, \ldots, \alpha_m$, and $\mathbf{e}_j$ are the eigenvectors of $\mathcal{I}_p(\hat{\boldsymbol{\theta}})$. Recall that the parameter vector $\boldsymbol{\theta}$ has m components. To obtain the SDP program for the query optimization when using the query criteria of Fisher information (FI) [Eq. (19)] in our AL strategy, we replace $\mathbf{e}_j$ by the eigenvectors of the identity matrix. In this case, $\mathbf{e}_j$ are m-dimensional canonical vectors with 1 in the jth component and zero elsewhere. The computational cost of solving the above SDP programs with a barrier interior-point method is $\mathcal{O}(n_{\mathcal{Q}}^2 m^3 + n_{\mathcal{Q}} m^4 + m^5)$ where we have denoted $n_{\mathcal{Q}} = |\mathcal{Q}|$ as the size of the query space of interest.

4. *Query advantage*

To compare resource requirements of different Hamiltonian learning (HL) methods for accomplishing a learning task, we introduce the concept of *query advantage*. The query advantage (QA) of an HL method in achieving a learning error of ϵ is

$$\text{QA} = 1 - \frac{\text{number of queries required by method}}{\text{number of queries required by baseline}}. \tag{24}$$

As discussed before, QA measures the amount of query reduction obtained by selecting a HL method over a baseline strategy. In this paper, we consider the passive learner equipped with an appropriately chosen estimation procedure as the baseline. We will specify the estimation procedure when discussing a QA result or it will be clear from context.

The benefits of quantifying QA for an HL method are twofold. Firstly, it allows us to comment on the performance boost obtained by using one HL method over another for accomplishing a learning task. Moreover, we can comment on learning tasks that can be achieved using one HL method but is unattainable by another. Secondly, it gives us a direct way to select a particular HL method based on minimal resources required, from a set of methods for a particular learning task by choosing the method with the highest QA.

As we will see in the next Sec. III, an active learning strategy is a framework for achieving query advantage and higher learning rates of convergence when possible.

5. *Related work*

Numerous methods exist for quantum Hamiltonian learning with different estimations and admissible experiments (also known as query space). This work is not intended to replace these Hamiltonian learning methods but propose a general framework for active learning (or adaptive experiment design) that can replace the step of proposing experiments within these methods.

Active learning for Hamiltonian learning has not received as much attention as has quantum state tomography and quantum process tomography. The only example we could find in literature is that of robust online Hamiltonian learning (ROHL) [37], which has since been used for scalable Bayesian estimation for Hamiltonian learning [29] and has been extended to Qinfer [38]. ROHL (or Qinfer) is a sequential active learner that proposes experiments (or queries) one shot at a time by minimizing a *risk* that is a function of the current estimate $\hat{\boldsymbol{\theta}}$ and examples seen so far. The proposed query for the $(k+1)$th shot then typically takes the form

$$x^{(k+1)} = \operatorname*{argmin}_{x\in\mathcal{Q}} R\big(X^{(k)}, Y^{(k)}; \hat{\boldsymbol{\theta}}^{(k)}\big), \tag{25}$$

where $X^{(k)}$ is the set of all queries taken so far, $Y^{(k)}$ are the associated measurement outcomes, and $\hat{\boldsymbol{\theta}}^{(k)}$ is the current estimate of the Hamiltonian parameters. In the case of ROHL and Qinfer, this is chosen to be Bayes risk given by

$$R(X^{(k)}, Y^{(k)}; \hat{\boldsymbol{\theta}}^{(k)}) = \mathbb{E}_{y|\hat{\boldsymbol{\theta}};\mathcal{Q}}[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(X^{(k)}, Y^{(k)}))^T$$
$$\times\, Q(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(X^{(k)}, Y^{(k)}))] \tag{26}$$

where the right-hand side can be interpreted as the expected posterior covariance matrix weighted against Q, which is a positive semidefinite matrix indicating the relative scale between the unknown parameters in $\boldsymbol{\theta}$. In practice, minimizing the above risk function is expensive and one may resort to different experimental design heuristics [66].

The main limitation of the above method is that it is very expensive to decide queries one at a time during learning. This will be true irrespective of our progress in quantum computers as this processing is done on a classical computer, and is especially an obstacle on current hardware given the usual limited access, and latencies within electronics. It is then desirable to batch up queries (or experiments), which has not been considered by earlier work. This paper, which proposes an active learner operating in batch mode thus fills this gap.

III. HAMILTONIAN ACTIVE LEARNING ALGORITHMS

We are now in a position to describe how to adapt probabilistic pool-based batch-mode active learning with query criteria of FI and FIR for Hamiltonian learning. The resulting algorithms are collectively called Hamiltonian active learning algorithms. We call HAL combined with the query criterion of Fisher information as HAL-FI and that with Fisher information ratio as HAL-FIR. We discuss how the resulting algorithms are expected to achieve query advantage over a specified baseline.

A. Algorithm

The HAL algorithm is summarized in Algorithm 1. We assume that the unknown Hamiltonian is time independent and a model parameterized by $\boldsymbol{\theta}^{\star}$ for the oracle (i.e., Hamiltonian with noise sources) is available to us. Inputs to the HAL algorithm include the initial query distribution $q^{(0)}$ to be used for sampling the initial set of $N_{\text{tot}}^{(0)}$ queries and the query optimization algorithm (QOA). When the query criterion is Fisher information (FI), the corresponding QOA is given by Algorithm 2, which solves Eq. (19). Similarly when the query criterion is Fisher information ratio (FIR), the corresponding QOA is given by Algorithm 3, which solves Eq. (21). Recall that the choice of query criteria and hence QOA depends on the learning task at hand: We consider FI for learning Hamiltonian parameters with low RMSE [Eq. (19)] and FIR for minimizing the testing error [Eq. (21)]. Additional inputs include the maximum number of batches $i_{\max}$ that will be issued during learning, which we will denote by N_b and the total experimental budget available.

Using the notation for batch-mode active learning as discussed in Sec. II C, we denote the batch of queries that are sampled at the ith round with respect to the query distribution $q^{(i)}$ as $X_q^{(i)}$ and the corresponding measurement outcomes from inputting these queries to the *oracle* as $Y_q^{(i)}$. The set of all queries made so far at any round is denoted by $X^{(i)}$ and their corresponding measurement outcomes as $Y^{(i)}$. We note that $|X_q^{(0)}| = |Y_q^{(0)}| = N_{\text{tot}}^{(0)}$, which is typically larger than $|X_q^{(i)}| = |Y_q^{(i)}| = N_b$, when prior information about the parameters is not available. The initial set of training examples is used to determine an initial value of the parameters, which is used for determining an informative albeit suboptimal query distribution $q^{(1)}$ through the given QOA. Note that in the QOA of Algorithms 2 and 3, we modify the query distribution $q^{(i)}$ obtained through solving Eq. (19) or (21), by mixing it with the uniform distribution over the query space p_U, i.e., $\mu q^{(i)} + (1-\mu)p_U$ where $0 \leqslant \mu \leqslant 1$ is the mixing coefficient. This is done to encourage exploration and is analogous to epsilon-greedy policies in reinforcement learning [67]. The value of μ typically depends on the number of queries made so far, and we set it to $\mu = 1 - 1/|X^{(i)}|^{1/6}$ as often used for such active learning algorithms [39,60].

Deviating from a vanilla batch model AL scheme, we require an additional input of the query space $\{\mathcal{Q}^{(i)}\}_{i\in[i_{\max}]}$ during training. We allow the query space to adaptively change from one batch to the next. How do we decide how the query space changes from one batch to the next? We remind ourselves that the query distribution in $\mathcal{I}_q$ is a joint probability distribution over the measurement operators, preparation operators and evolution times. Conditioned on a particular evolution time, we would not expect changing $\mathcal{M}$ or $\mathcal{U}$ to help in reducing the query complexity. The query space is then adaptively grown by growing $\mathcal{T}$ linearly or exponentially with each batch. We note that $\mathcal{Q}^{(1)} \subset \mathcal{Q}^{(2)} \subset \ldots \subset \mathcal{Q}^{(i_{\max})}$.

The output of the algorithm is an estimate $\hat{\boldsymbol{\theta}}$ of the true Hamiltonian model parameters $\boldsymbol{\theta}^{\star}$ learned through active learning using N queries. In the HAL algorithm as presented in Algorithm 1, we use the maximum likelihood estimator in the HAL algorithm but this may be replaced by other estimation methods such as regression or Bayesian estimation as we show later in Sec. V. HAL-FI produces an estimate $\hat{\boldsymbol{\theta}}$ that has an RMSE lower than what would be obtained without any active learning using a budget of N queries, and batches of size N_b. Similarly, HAL-FIR produces an estimate $\hat{\boldsymbol{\theta}}$ that performs well in prediction of queries to the Hamiltonian against the testing distribution p_{test}.

Algorithm 1. Hamiltonian active learning (HAL).

```
Input: Initial number of queries $N_{\text{tot}}^{(0)}$, Batch size $N_b$, Initial query distribution $q^{(0)}$, Maximum number of batches $i_{\max}$, Adaptively growing query space $\{\mathcal{Q}^{(i)}\}_{i\in[i_{\max}]}$, oracle, query optimization algorithm (QOA)
Output: $\hat{\boldsymbol{\theta}}$
Sample $N_{\text{tot}}^{(0)}$ queries $X_q^{(0)}$ from $\mathcal{Q}^{(0)}$ according to $q^{(0)}$
Obtain measurement outcomes $Y_q^{(0)}$ by sending queries $X^{(0)}$ to oracle
Set $X^{(0)} = X_q^{(0)}$ and $Y^{(0)} = Y_q^{(0)}$
Compute MLE estimate: $\hat{\boldsymbol{\theta}}^{(i-1)} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; X^{(0)}, Y^{(0)})$
for $i = 1 : i_{\max}$ do
    Solve $q^{(i)}$ through QOA (Algorithm 2 or 3)
    Sample $N_b$ queries $X_q^{(i)}$ from $\mathcal{Q}^{(i)}$ w.p. $q^{(i)}$
    Update number of queries: $N_{\text{tot}}^{(i)} = N_{\text{tot}}^{(i-1)} + N_b$
    Obtain measurement outcomes $Y^{(i)}$ by issuing queries $X^{(i)}$ to oracle
    Set $X^{(i)} = X^{(i-1)} \bigcup X_q^{(i)}$ and $Y^{(i)} = Y^{(i-1)} \bigcup Y_q^{(i)}$
    Compute MLE estimate: $\hat{\boldsymbol{\theta}}^{(i)} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; X^{(i)}, Y^{(i)})$
return $\hat{\boldsymbol{\theta}}^{(i_{\max})}$
```

B. Comment on query advantage and Heisenberg limited scaling

In this section, we provide intuition as to why we expect HAL-FI to provide query advantage and when Heisenberg limited scaling may be expected in experiment. The intuition and claims made here are backed by empirical evidence in Sec. V.

We start by comparing the performance of a passive learner L_{PL} and an active learner L_{HAL}, which uses HAL-FI. We

Algorithm 2. Query optimization based on Fisher information (FI).

```
Input: Number of queries made so far $N_{\text{tot}}^{(i-1)}$, Batch size $N_b$,
Query space $\mathcal{Q}^{(i)}$, Current parameter estimates $\hat{\boldsymbol{\theta}}^{(i)}$
Output: $q^{(i)}$
Set $N_{\text{tot}}^{(i)} = N_{\text{tot}}^{(i-1)} + N_b$
Solve $q^{(i)} = \arg\min_q \operatorname{Tr}(\mathcal{I}_q^{-1}(\hat{\boldsymbol{\theta}}^{(i)}))$ subject to $\sum_{x\in\mathcal{Q}^{(i)}} q(x) = 1$, and $0 \leqslant q(x) \leqslant 1, \forall x \in \mathcal{Q}^{(i)}$
Obtain uniform distribution over query space: $p_U = 1/|\mathcal{Q}^{(i)}|$
Set mixing coefficient: $\mu = 1 - 1/|N_{\text{tot}}^{(i-1)}|^{1/6}$
Modify query distribution: $q^{(i)} = \mu q^{(i)} + (1-\mu)p_U$
return $q^{(i)}$
```

Algorithm 3. Query optimization based on Fisher information ratio (FIR).

```
Input: Number of queries made so far $N_{\text{tot}}^{(i-1)}$, Batch size $N_b$,
Query space $\mathcal{Q}^{(i)}$, Current parameter estimates $\hat{\boldsymbol{\theta}}^{(i)}$, Testing Distribution $p_{\text{test}}$
Output: $q^{(i)}$
Set $N_{\text{tot}}^{(i)} = N_{\text{tot}}^{(i-1)} + N_b$
Compute model fisher information corresponding to $p_{\text{test}}$: $\mathcal{I}_{p_{\text{test}}}(\hat{\boldsymbol{\theta}})$
Solve $q^{(i)} = \arg\min_q \operatorname{Tr}(\mathcal{I}_q^{-1}(\hat{\boldsymbol{\theta}}^{(i)})\mathcal{I}_{p_{\text{test}}}(\hat{\boldsymbol{\theta}}^{(i)}))$ subject to $\sum_{x\in\mathcal{Q}^{(i)}} q(x) = 1$, and $0 \leqslant q(x) \leqslant 1, \forall x \in \mathcal{Q}^{(i)}$
Obtain uniform distribution over query space: $p_U = 1/|\mathcal{Q}^{(i)}|$
Set mixing coefficient: $\mu = 1 - 1/|N_{\text{tot}}^{(i-1)}|^{1/6}$
Modify query distribution: $q^{(i)} = \mu q^{(i)} + (1-\mu)p_U$
return $q^{(i)}$
```

assume that both of these learners use the same estimation method to estimate $\hat{\boldsymbol{\theta}}$ from the training examples generated. Consider the second round of learning ($i = 2$) with a fixed query space $\mathcal{Q}$ and where both learners have seen the same data $D^{(1)}$ so far. L_{PL} will uses queries uniformly sampled from $\mathcal{Q}$ and L_{HAL} will use HAL-FI to decide the query distribution to use for sampling from $\mathcal{Q}$. The corresponding Fisher information is $\mathcal{I}_{p_U}(\hat{\boldsymbol{\theta}}^{(1)})$ (where p_U is the uniform distribution) and $\mathcal{I}_q(\hat{\boldsymbol{\theta}}^{(1)})$. From Eq. (19), we know that $\operatorname{Tr}(\mathcal{I}_q^{-1}(\hat{\boldsymbol{\theta}}^{(1)})) \leqslant \operatorname{Tr}(\mathcal{I}_{p_U}^{-1}(\hat{\boldsymbol{\theta}}^{(1)}))$. We thus expect L_{HAL} to require the fraction of $r^{(1)} \approx \operatorname{Tr}(\mathcal{I}_q^{-1}(\hat{\boldsymbol{\theta}}^{(1)}))/\operatorname{Tr}(\mathcal{I}_{p_U}^{-1}(\hat{\boldsymbol{\theta}}^{(1)})) \leqslant 1$ number of queries compared to L_{PL} to achieve the same variance reduction. This is compounded through the following learning rounds, leading to an overall reduction in queries required (or gain in query advantage) by L_{HAL} to reach the same RMSE compared to L_{PL}. Moreover, in latter rounds ($i > 2$), the fraction of queries required by L_{HAL} may further decrease due to more informative training examples available to obtain $\hat{\boldsymbol{\theta}}^{(i)}$, which is then used to obtain the query distribution.

We claim that HAL-FI with an appropriately chosen adaptively growing query space by adaptively growing $\mathcal{T}$ can achieve Heisenberg limited scaling in Hamiltonian parameters where possible. If the query space is not *rich* enough, it will not be possible to determine a sequence of queries to achieve Heisenberg limited scaling. Without an adaptively growing query space (i.e., with fixed query space), the scaling of the number of queries N with error parameter ϵ is $O(1/\epsilon^2)$ as is dictated by the Cramer-Rao bound. The query complexity by using an active learner only improves by a constant factor in such a case. If one chose to adaptively grow the query space during training without an active learning strategy and chose for example an uniform distribution over the new query space (i.e., carry out passive learning), Heisenberg limited scaling would not be expected as it would become exponentially more unlikely to sample an informative query. Moreover, Heisenberg limited scaling is expected as long as the evolution time is below the decoherence time. Beyond the decoherence time, the oracle will start losing its quantum behavior.

If the so chosen adaptively growing query space cannot be used to achieve Heisenberg limited scaling, it should still be possible to achieve Heisenberg limited scaling for a subset of the Hamiltonian parameters provided that the goal is to now learn these parameters and we are provided information about the other parameters. This learning task often occurs in practice during recalibrations of quantum devices when it is required to learn a subset of parameters that are known to fluctuate significantly with time but information about the other parameters can be used from previous calibrations during estimation. We provide empirical evidence for this claim in Sec. V where we consider the application of the HAL-FI to cross-resonance Hamiltonians. The results of HAL-FI for this particular application are summarized in Table I.

What distinguishes HAL-FI from other methods such as Floquet calibration [28], which have been shown to achieve Heisenberg limited scaling is that it does not require prior specification of experiments and their order of implementation. This is decided by the HAL-FI during learning. Moreover, HAL-FI utilizes single-shot outimes from queries, instead of requiring expectation values. In practice, this can result in multiple orders of magnitude reduction in queries required. Finally, HAL-FI can achieve query advantage over passive learners for complete query spaces even when Heisenberg limited scaling cannot be achieved.

C. Computational cost and extensions

A consequence of adaptively growing the query space over rounds during learning is that the SDP programs [Eq. (23)] corresponding to the query optimizations of Eqs. (19) and (21) [with computational cost $\mathcal{O}(n_{\mathcal{Q}}^2 m^3 + n_{\mathcal{Q}} m^4 + m^5)$] become increasingly more computationally expensive to solve over rounds. If $n_{\mathcal{Q}}$ grows exponentially, each iteration of the query optimization problem becomes more exponentially expensive

TABLE I. Query complexity of HAL-FI for Hamiltonian learning (Problem II A 3 a) under different conditions as observed on a real quantum device. The case of fixed query space corresponds to adaptivity in query space being *none*.

Adaptivity in query space	Scaling of N	Scaling of N (recalibration)
None	$O(1/\epsilon^2)$	$O(1/\epsilon^2)$
Linearly growing $\mathcal{T}$	$O(1/\epsilon^2)$	$O(1/\epsilon^{2/3})$
Exponentially growing $\mathcal{T}$	$O(1/\epsilon^2)$	$> O(1/\epsilon)$

to solve. This can be circumvented by reducing the number of queries to optimize over using uncertainty filtering [68], thereby effectively reducing the size of the query space n_Q over which the query optimization is carried out. Uncertainty filtering for our application of Hamiltonian learning to the cross-resonance Hamiltonian is discussed in Appendix B 2.

The HAL-FI and HAL-FIR algorithms can be generalized to different experimental setups or requirements. The HAL algorithm presented in Algorithm 1 uses the stopping criterion of maximum number of batches of queries issued during learning but other stopping criteria such as the ℓ_2 norm of the differences in consecutive parameter values $||\hat{\boldsymbol{\theta}}^{(i)} - \hat{\boldsymbol{\theta}}^{(i-1)}||_2$ could also be used.

IV. HAMILTONIAN LEARNING FOR A TWO-QUBIT SUPERCONDUCTING CR GATE: MODEL AND SETUP

To assess the performance of the HAL algorithms described in Sec. III over a passive learner and empirically verify our claims, we consider the application of learning cross-resonance (CR) Hamiltonians on superconducting IBM quantum devices. In this section, we describe the model of the two-qubit cross-resonance Hamiltonian, and the set of queries that we can make to it. This is followed by a description of the different noise sources that affect the quantum system and the resulting likelihood of measurement outcomes given queries. The evaluation of the likelihood is required by all of our estimators (Sec. II A 4). In turn, we then describe different estimators used for Hamiltonian learning including MLE and a Bayesian estimator. Finally, we describe our implementation of the HAL algorithms in the context of numerical experiments for learning the CR Hamiltonians. A description of the IBM quantum devices employed for assessing the performance of the HAL algorithms can be found in Appendix A.

A. Cross-resonance Hamiltonian

The cross resonance (CR) gate is a two-qubit entangling gate for superconducting qubits requiring only microwave control, which allows for the use of fixed-frequency transmon qubits [69,70]. Using appropriate pulse sequences such as multipulse echos and cancellation tones [71], the CR gate can be transformed to a locally equivalent CNOT gate [72]. Combined with arbitrary single qubit gates, this then forms a complete set of gates for universal quantum computation.

The Hamiltonian of the cross-resonance (CR) gate has the following structure:

$$H_{CR} = \frac{\sigma_Z \otimes A}{2} + \frac{\sigma_I \otimes B}{2}, \tag{27}$$

$$A = c_{ZI}\sigma_I + c_{ZX}\sigma_X + c_{ZY}\sigma_Y + c_{ZZ}\sigma_Z, \tag{28}$$

$$B = c_{IX}\sigma_X + c_{IY}\sigma_Y + c_{IZ}\sigma_Z, \tag{29}$$

where $\{\sigma_I, \sigma_X, \sigma_Y, \sigma_Z\}$ are the single-qubit Paulis and $c_{ab} \in \mathbb{R}$ are real coefficients of the Pauli product terms $\sigma_a \otimes \sigma_b$. The above time-independent Hamiltonian description of the cross-resonance gate can be obtained through theoretical models based on effective block-diagonal Hamiltonian techniques [73]. In our experiments, which we describe in Sec. IV B, the CR gate is implemented without using an echo [71] to refocus the $\sigma_I\sigma_X$, $\sigma_Z\sigma_Z$, and $\sigma_Z\sigma_I$ terms. However, we measure only the target qubit through our queries and thus effectively neglect the $\sigma_Z\sigma_I$ term. The target qubit is typically chosen to be qubit 1 in a (0,1) qubit pair and is specified for the different quantum devices we consider in Appendix A 1. The effective removal of the $\sigma_Z\sigma_I$ term from Eq. (29) results in the following simplified CR Hamiltonian, which we consider throughout the rest of our paper,

$$H = \sum_{\substack{a\in\{I,Z\} \\ b\in\{X,Y,Z\}}} J_{ab}\sigma_a \otimes \sigma_b \tag{30}$$

where we have used the parameter vector $\mathbf{J} = [J_{IX}, J_{IY}, J_{IZ}, J_{ZX}, J_{ZY}, J_{ZZ}]^T$ to denote the nonzero coefficients of the corresponding Pauli product terms and have omitted the subscript CR. Learning the unknown Hamiltonian of the two-qubit CR gate is then reduced to estimating the unknown parameter vector $\mathbf{J}$.

Noting the block-diagonal structure of H, we can express it in the usual computational basis of $(|00\rangle, |01\rangle, |10\rangle, |11\rangle)$ as

$$H = \begin{bmatrix} a_0 & \beta_0^\star & 0 & 0 \\ \beta_0 & -a_0 & 0 & 0 \\ 0 & 0 & a_1 & \beta_1^\star \\ 0 & 0 & \beta_1 & -a_1 \end{bmatrix}, \text{ where}$$

$$\begin{aligned} a_j &= J_{IZ} + (-1)^j J_{ZZ} \\ \beta_j &= (J_{IX} + (-1)^j J_{ZX}) + i(J_{IY} + (-1)^j J_{ZY}) \end{aligned} \tag{31}$$

where the subscript $j \in \{0, 1\}$ is used to refer to the two different blocks. The two blocks have similar structure with their elements differing in the sign of J_{ZX}, J_{ZY}, and J_{ZZ}. Using a_j, and β_j for $j \in \{0, 1\}$ from Eq. (31), we define the following parameters:

$$\omega_j = \sqrt{a_j^2 + |\beta_j|^2}, \quad \delta_j = \sin^{-1}\frac{a_j}{\omega_j}, \quad \phi_j = \arg(\beta_j) \tag{32}$$

where the subscript $j = 0$ is used to denote the preparation operator $U = \sigma_I\sigma_I$ and $j = 1$ to denote $U = \sigma_X\sigma_I$. We then arrive at an alternate parametrization (related to the spectral decomposition of H), which turns out to be useful for simplifying expressions for probability, likelihood and Fisher information: $\mathbf{\Lambda} = (\omega_0, \delta_0, \phi_0, \omega_1, \delta_1, \phi_1)^T$. The unitary operator $U(t) = e^{-iHt}$ in the usual computational basis is then given by

$$U(t) = \begin{bmatrix} \cos(\omega_0 t) - i\sin(\delta_0)\sin(\omega_0 t) & -ie^{-i\phi_0}\cos(\delta_0)\sin(\omega_0 t) & 0 & 0 \\ -ie^{i\phi_0}\cos(\delta_0)\sin(\omega_0 t) & \cos(\omega_0 t) + i\sin(\delta_0)\sin(\omega_0 t) & 0 & 0 \\ 0 & 0 & \cos(\omega_1 t) - i\sin(\delta_1)\sin(\omega_1 t) & -ie^{-i\phi_1}\cos(\delta_1)\sin(\omega_1 t) \\ 0 & 0 & -ie^{i\phi_1}\cos(\delta_1)\sin(\omega_1 t) & \cos(\omega_1 t) + i\sin(\delta_1)\sin(\omega_1 t) \end{bmatrix} \tag{33}$$

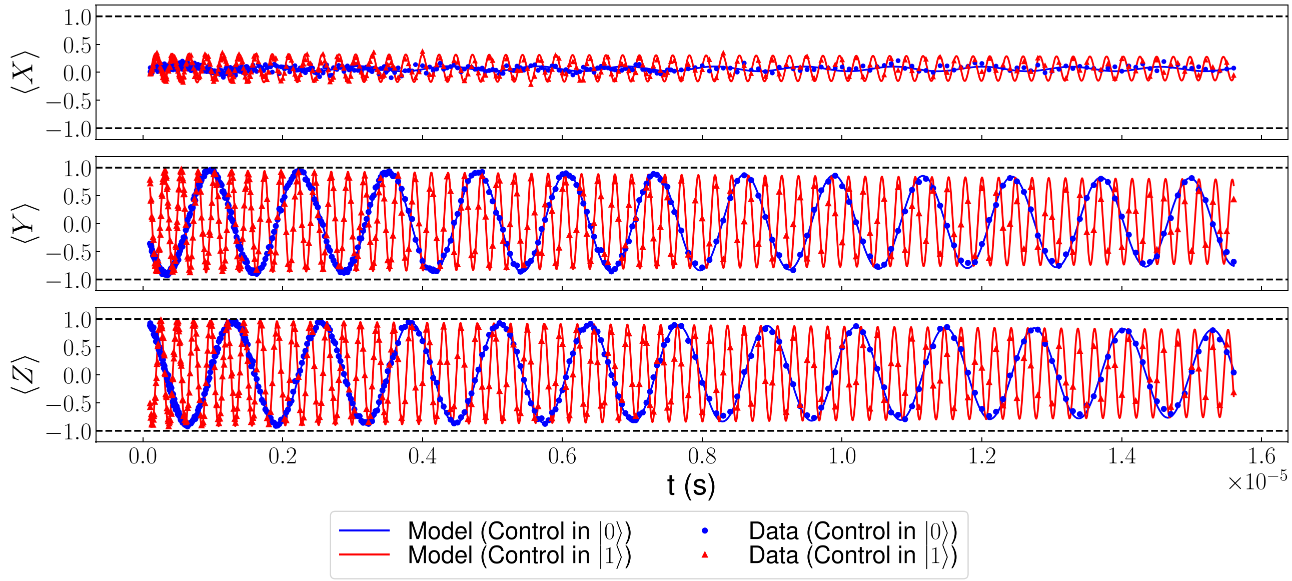

FIG. 3. Rabi oscillations obtained experimentally through queries to a CR Hamiltonian on the IBM quantum device (*ibmq_boeblingen*) for different measurement operators (rows), preparation operators (markers), and evolution times t (x axis). The set of measurement and preparation operators will be described in Sec. IV B 2. The experimental data is indicated by markers and the model fit to the data by lines. The model fit was generated by learning the CR Hamiltonian [Eq. (31)] from experimental data using MLE and predicting values of Rabi oscillations for the query space using the learned Hamiltonian.

where we have used the parameter vector of $\mathbf{\Lambda}$ defined in Eq. (32). Moreover, the different components of $\mathbf{\Lambda}$ can be bounded based on their definitions and these bounds will help us later while solving the MLE problems. By construction, we have $-\frac{\pi}{2} \leqslant \delta_{0,1} \leqslant \frac{\pi}{2}$ and $\phi_{0,1}$ can be bounded within any interval of size 2π, e.g., $-\pi \leqslant \phi_{0,1} \leqslant \pi$. Assuming δt is the average time increment between distinct ordered values of evolution times $t \in \mathcal{T}$, $\omega_{0,1}$ can be bounded based on the Nyquist sampling theorem as $0 \leqslant \omega_{0,1} \leqslant \frac{\pi}{\delta t}$.

To obtain the physical meaning of the parameter vector $\mathbf{\Lambda}$, we consider Rabi oscillations. For different measurement operators $M \in \mathcal{M}$ and preparation operators $U \in \mathcal{U}$, the corresponding Rabi oscillation is the difference in probability densities of the ground state and excited state of the target qubit with time $t \in \mathcal{T}$. A typical example of the Rabi oscillations from querying the CR Hamiltonian on a noisy quantum system is shown in Fig. 3. We remark that $\omega_{0,1}$ defines the frequency of the Rabi oscillations for the two different preparation operators we consider. The parameters $\delta_{0,1}$ and $\phi_{0,1}$ define the offsets, amplitudes, and phase shifts of the Rabi oscillations. In Fig. 3, we can see the effects of different noise sources such as readout and depolarization, which will be discussed in Sec. IV C.

B. Experimental setup

In this section, we give a quick overview of the different IBM quantum devices that we employ for our application of Hamiltonian learning. This is followed by a description of the query space considered for our application.

1. Quantum devices

We will present data and results for four different IBM quantum devices, which we call A, B, C, and D. All of these devices are based on superconducting architectures requiring only microwave control [69,70] and consist of fixed-frequency transmon qubits with shared quantum buses [74]. Device A is a two-qubit device. Device B is a four-qubit device on which we will query only one of the CR gates that can be implemented between two qubits. Device C is a five-qubit device with a bow-tie layout. Device D is the 20-qubit *ibmq_boeblingen*, which was accessed via the IBM quantum cloud computing service. See Fig. 16 in Appendix A 1 for the connectivity maps of all the devices. We give a summary of the properties of the pairs of qubits involved in the cross-resonance Hamiltonians we considered on these devices in Table VII in Appendix A 1.

2. Query space

To prepare the initial state, we consider the set of preparation operators $\mathcal{U} = \{\sigma_I \otimes \sigma_I, \sigma_X \otimes \sigma_I\}$ applied to the pure state $|00\rangle$. Assuming the first (left) qubit is the control and the second (right) qubit is the target, the effect of the preparation operators is to place the control in $|0\rangle$ and $|1\rangle$ respectively. We evolve the initial state $|\psi(0)\rangle$ for time $t \in \mathcal{T}$, which we will specify when discussing the results of our application of Hamiltonian learning in Sec. V. Finally after obtaining the final state $|\psi(t)\rangle$, we apply the measurement operators in $\mathcal{M} = \{\sigma_I \otimes \exp(i\frac{\pi}{4}\sigma_Y), \sigma_I \otimes \exp(-i\frac{\pi}{4}\sigma_X), \sigma_I \otimes \sigma_I\}$ and measure only the second qubit, which we have chosen as the target qubit. The query space is then $\mathcal{Q} = \mathcal{M} \times \mathcal{U} \times \mathcal{T}$.

Queries to the CR Hamiltonians between different qubit pairs on the IBM quantum devices are made through appropriate pulse sequences. These pulse sequences are constructed and executed on the hardware using `Qiskit-Pulse` [75], which is a pulse programming module within `Qiskit` [76].

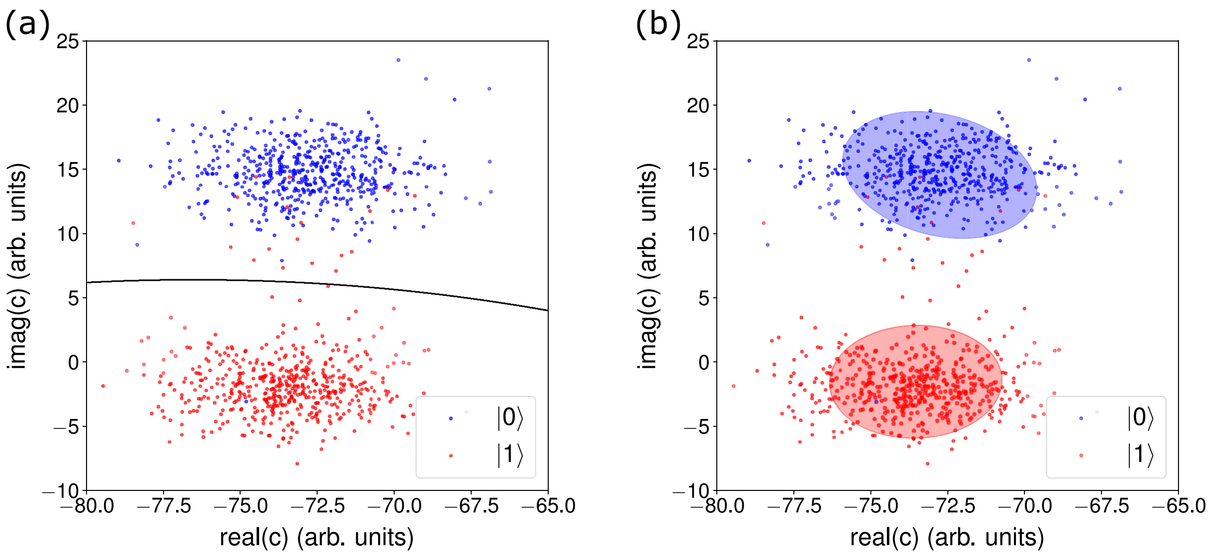

FIG. 4. Characterization of readout noise from calibration data of single qubit readouts considering a bit-flip channel model using (a) a trained Bayesian naive classifier and (b) fitting Gaussian distributions. In (a) and (b), the experimental data points of the complex readout signal **c** are shown as markers. In (a), the decision boundary is shown as a line. In (b), the contours indicate the single standard deviation.

Description of how a query to the CR gate on an IBM quantum device is specified is given in Appendix A 2.

In our experimental setup, we obtain measurements of the single-shot signal (integrated cavity amplitude) **c** from the IBM quantum devices, which is a function of the measurement outcomes **y**, which we have described earlier. In the following Sec. IV C, we will discuss how to model the different noise sources that affect our system and how they are determined through experiments.

C. Estimates of noise and nonidealities for experimental system

Using the noise models presented in Sec. II B, we give the specific relevant models for readout noise, imperfect pulse shaping, and decoherence for the different IBM quantum devices, which we study.

Let us introduce some notation that we will use in the following discussions. We use the index k for the queries. The kth query is given by $x^{(k)} = (M^{(k)}, U^{(k)}, t^{(k)})$, and the corresponding measurement outcome of the target qubit as $y^{(k)}$. The measurement outcome $y^{(k)}$ is not directly observed but inferred from the corresponding signal $\mathbf{c}^{(k)}$. We denote the inferred value as $\hat{y}^{(k)}$.

1. Readout noise

As discussed in Sec. II B, we assume the measurement noise model to be a bit-flip channel with the input of unobserved measurement outcomes y and the output of readout $\tilde{y}$ observed through signal **c**. For this inference task, we use calibration data of single qubits initially prepared in states $|0\rangle$ or $|1\rangle$, and subsequently measured in the usual computational basis. In Fig. 4(a), we plot different realizations $\{y^{(k)}, \mathbf{c}^{(k)}\}_{k=1}^{100}$ from IBM quantum device D *ibmq_boeblingen* that were used for training a binary classifier. The classifier provides us with the ability to predict y given **c**. Moreover, the misclassification errors $p_{\hat{\mathrm{y}}|\mathrm{y}}(1|0)$ and $p_{\hat{\mathrm{y}}|\mathrm{y}}(0|1)$ can be used to approximate the properties of the bit-flip channel, in particular the conditional probabilities of a bit-flip $p_{\tilde{\mathrm{y}}|\mathrm{y}}(1|0)$ and $p_{\tilde{\mathrm{y}}|\mathrm{y}}(0|1)$ respectively. As these are independent of values of **c** here, we denote the conditional probabilities of a bit-flip as $r_0 = p_{\tilde{\mathrm{y}}|\mathrm{y}}(1|0)$ and $r_1 = p_{\tilde{\mathrm{y}}|\mathrm{y}}(0|1)$. The MLE of the parameters incorporating this noise model is given by

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -\frac{1}{N}\sum_{k=1}^{N} \log[p_{\tilde{\mathrm{y}}|\mathrm{x}}(\tilde{y}^{(k)}|x^{(k)};\boldsymbol{\theta})] \tag{34}$$

$$= \arg\min_{\boldsymbol{\theta}} -\frac{1}{N}\sum_{k=1}^{N} \log[(1-\hat{y}^{(k)})((1-r_0)p_{\mathrm{y}|\mathrm{x}}(y^{(k)}=0|x^{(k)};\boldsymbol{\theta}) + r_1(1-p_{\mathrm{y}|\mathrm{x}}(y^{(k)}=1|\boldsymbol{x}^{(k)};\boldsymbol{\theta})))\tag{35}$$

$$+\hat{y}^{(k)}((1-r_1)(1-p_{\mathrm{y}|\mathrm{x}}(y^{(k)}=1|x^{(k)};\boldsymbol{\theta})) + r_0 p_{\mathrm{y}|\mathrm{x}}(y^{(k)}=0|x^{(k)};\boldsymbol{\theta}))]\tag{36}$$

where $p_{\mathrm{y}|\mathrm{x}}(y^{(k)}|x^{(k)};\boldsymbol{\theta})$ is given by Eq. (1). Alternately, instead of assigning a deterministic result $\hat{y}^{(k)}$ for each $c^{(k)}$, we can incorporate $p_{\mathrm{c}|\mathrm{y}}$ directly into the log-likelihood function, which could yield a more accurate model. This could

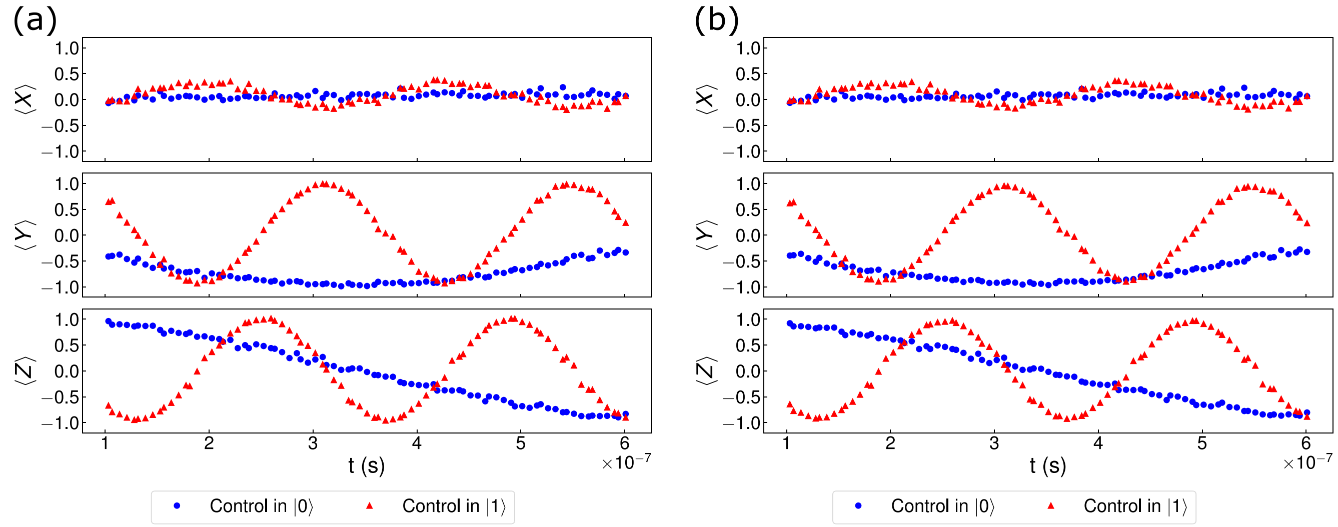

FIG. 5. Examples of Rabi oscillations computed from experimental data collected from IBM quantum device D *ibmq_boeblingen* for different measurement operators M (rows), preparation operators U (markers), and evolution times t (x axis). These were computed (a) using a binary classifier and (b) using Gaussian distribution fits. The difference between the two approaches is negligible when the misclassification error of the binary classifier is low (r_0, r_1 < 0.01) as is the case here.

be done through our choice of binary classifier or by fitting a distribution to the training data. Noting that single qubit measurement outcomes correspond to their energy levels, we fit Gaussian distributions to the training data and hence obtain a parameteric form of $p_{\mathrm{c|y}}$ in Fig. 4(b). The MLE is now

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -\frac{1}{N}\sum_{k=1}^{N} \log[p_{\mathrm{c|y}}(c^{(k)}|0)p_{\mathrm{y|x}}(y^{(k)}=0|x^{(k)};\boldsymbol{\theta}) + p_{\mathrm{c|y}}(c^{(k)}|1)(1-p_{\mathrm{y|x}}(y^{(k)}=0|x^{(k)};\boldsymbol{\theta}))]. \tag{37}$$

A useful tool for calibration and diagnostics is Rabi oscillations, which we denote by $p_{\mathrm{rabi}}(x)$. Rabi oscillations are obtained from evaluating the difference in the population densities of the ground state and excited state of the target qubit or $p_{\mathrm{rabi}}(x) = p_{\mathrm{y|x}}(0|x) - p_{\mathrm{y|x}}(1|x)$. We can compute Rabi oscillations in two different ways, either through binary classification or through fitting Gaussians. Using the misclassification errors from the binary classifier, we can then write

$$\hat{p}_{\mathrm{rabi}}(x) = p_{\hat{\mathrm{y}}|\mathrm{x}}(0|x)\left(\frac{1-r_0+r_1}{1-r_0-r_1}\right) - p_{\hat{\mathrm{y}}|\mathrm{x}}(1|x)\left(\frac{1+r_0-r_1}{1-r_0-r_1}\right), \tag{38}$$

where we have denoted the computed Rabi oscillations as $\hat{p}_{\mathrm{rabi}}(x)$. While $p_{\mathrm{y|x}}(0|x)$ and $p_{\mathrm{y|x}}(1|x)$ are guaranteed to be valid probability distributions, the above estimation does not ensure that $\hat{p}_{\mathrm{rabi}}(x)$ is bounded by -1 and 1. To obtain more accurate estimates of the Rabi oscillations, we can solve the following MLE problem:

$$\hat{p}_{\mathrm{rabi}}(x) = \arg\min_{q\in[-1,1]}\left(-\sum_{k} \mathbb{1}\{x^{(k)}=x\}\log[(1+q)p_{\mathrm{c|y}}(c^{(k)}|0) + (1-q)p_{\mathrm{c|y}}(c^{(k)}|1)]\right), \tag{39}$$

where we use the estimated conditional distributions $p_{\mathrm{c|y}}$ from the Gaussian fits. The indicator function $\mathbb{1}\{x^{(k)}=x\}$ is used to ensure that the summation is only over measurement outcomes of given query x. We can write down the analytical expressions for $p_{\mathrm{rabi}}(x)$ for each measurement operator M and preparation operator U noting that $x=(M,U,t)$ as

$$M_{\langle X\rangle} : p_{\mathrm{rabi}}(x) = \sin\delta_j\cos\delta_j\cos\phi_j + \cos\delta_j\sqrt{1-\cos^2\delta_j\cos^2\phi_j}\cos(2\omega_j t+\alpha_j), \tag{40}$$

$$M_{\langle Y\rangle} : p_{\mathrm{rabi}}(x) = \sin\delta_j\cos\delta_j\sin\phi_j + \cos\delta_j\sqrt{1-\cos^2\delta_j\sin^2\phi_j}\cos(2\omega_j t+\gamma_j), \tag{41}$$

$$M_{\langle Z\rangle} : p_{\mathrm{rabi}}(x) = 1-2\cos^2\delta_j\sin^2(\omega_j t) = \sin^2\delta_j + \cos^2\delta_j\cos(2\omega_j t), \tag{42}$$

where we denote $\alpha_j = \arg((-\sin\delta_j\cos\phi_j) + i(-\sin\phi_j))$ and $\gamma_j = \arg((-\sin\delta_j\sin\phi_j) + i\cos\phi_j)$ in the above expressions. The subscript $j=0$ is used to denote the preparation operator $U = \sigma_I\sigma_I$ and $j=1$ to denote $U = \sigma_X\sigma_I$.

In Fig. 5, we plot the Rabi oscillations for the different $M \in \mathcal{M}$ and $U \in \mathcal{U}$ over the time range of $t \in \mathcal{T}$ using the above two methods. We observe that the Rabi oscillations are not bounded between -1 and $+1$ for the case of $M_{\langle Y \rangle}$ in Fig. 5(a) when using misclassification error to compensate for the readout noise. The estimated Rabi oscillations in Fig. 5(b) do not suffer from the same issue.

As we will see later in Sec. IV D, Rabi oscillations will also be used in our estimation procedure for obtaining an initial guess for the parameter estimate $\hat{\boldsymbol{\theta}}$ that is used as an input to the optimizer for solving the MLE problem. It can also be used as a quantitative tool for ascertaining how well the model fits the data. This will become apparent in the next few sections where we discuss other noise models.

2. Imperfect pulse shaping

Another nonideality is introduced through the pulses used to control the Hamiltonian and implement different unitary operators. It is convenient to think of cross-resonance control pulses as rectangular pulses that modulate the sinusoidal control pulse. The modulated signal results in unitary operators of the form $U(t_1) = \exp(-iHt_1)$, which we would want to implement in a quantum circuit. However, in practice, rectangular pulses cause significant amounts of signal energy to be distributed above and below the frequency of the control sinusoid. This distribution of energy can potentially excite higher energy states of the superconducting transmon being used as a qubit (i.e., $|2\rangle$ and above) as well as excite neighboring spectator transmon qubits. To minimize such effects, pulse shaping is employed to reduce this energy spread by smoothing the rising and falling edges of the pulse, which has the effect of reducing the magnitudes of the frequency artifacts above and below the frequency of the control sinusoid.

Pulse shaping is accomplished by taking a Gaussian-shaped pulse, splitting it in half, and then inserting a rectangular pulse between the halves. This results in the `GaussianSquare` pulse, described in further detail in Appendix A 2. Thus, we actually implement operators of the form $\tilde{U}(t_1) = \mathbb{T}\exp\left(-i\int_0^{t_1}\tilde{H}(t;\boldsymbol{\theta})dt\right)$ where $\mathbb{T}$ is the time ordering operator and $\tilde{H}(t;\boldsymbol{\theta})$ is the cross-resonance Hamiltonian due to imperfect control at any particular time t with parameters $\boldsymbol{\theta}$. Up to a first-order approximation in t, we can model $\tilde{H}(t;\boldsymbol{\theta})$ as (see Appendix A 3 for details)

$$\tilde{H}(t;\boldsymbol{\theta}) = v(t)H(\boldsymbol{\theta}), \tag{43}$$

where $H(\boldsymbol{\theta})$ is the time-independent cross-resonance Hamiltonian and $v(t)$ is a function of the cross-resonance pulse amplitude. Using this, we can now derive an expression for the unitary of Hamiltonian evolution. We denote Δt_r and Δt_f as the time durations of the rising and falling edges of the shaped pulse. The central portion of $v(t')$ is a rectangular function; i.e., for $\Delta t_r \leqslant t' \leqslant t - \Delta t_f$, $v(t') = \mathbb{1}_{t' \in [\Delta t_r, t-\Delta t_f]}$ where t is the duration of the pulse. We then have

$$\tilde{U}(t) = \exp\left(-iH(\boldsymbol{\theta})\int_0^t v(t')dt'\right) = \exp\left(-iH(\boldsymbol{\theta})\left[(t - \Delta t_f - \Delta t_r) + \underbrace{\int_0^{\Delta t_r} v(t')dt' + \int_{t-\Delta t_f}^{\Delta t_f} v(t')dt'}_{\Delta t_{\text{eff}}}\right]\right) \tag{44}$$

$$= \exp\left(-iH(\boldsymbol{\theta})(t_{\text{expt}} + \Delta t_{\text{eff}})\right), \tag{45}$$

where in the last step, we set $t_{\text{expt}} = t - \Delta t_f - \Delta t_r$ which is the evolution time that is reported in our experiments. It should be noted that this can change from one experimental setup to another. The value of Δt_{eff} can be interpreted as an effective total edge duration that takes the shapes of the rising and falling edges into account. This first-order pulse-shaping model introduces another model parameter, namely Δt_{eff}, that we need to estimate. We can do this directly in our MLE,

$$[\hat{\boldsymbol{\theta}}, \Delta\hat{t}_{\text{eff}}] = \arg\min_{\boldsymbol{\theta}, \Delta t_{\text{eff}}} -\frac{1}{N}\sum_{k=1}^{N}\log\left[\sum_{y\in\{0,1\}}\left(p_{\text{c}|\text{y}}(c^{(k)}|y)\sum_{z\in\{0,1\}}|\langle yz|M^{(k)}e^{-iH(\boldsymbol{\theta})(t^{(k)}+\Delta t_{\text{eff}})}U^{(k)}|00\rangle|^2\right)\right]. \tag{46}$$

While this can be certainly done while learning the Hamiltonian, one can also determine the dependence of Δt_{eff} on the different Hamiltonian parameters using prior calibration data. We determined that Δt_{eff} depends only on the parameters of $\omega_{0,1}$ as $\Delta\hat{t}_{\text{eff}}(\boldsymbol{\theta}) = a/(\omega + b\omega^2)$. In Fig. 6, we plot the dependence of Δt (short for Δt_{eff}) on ω for the IBM quantum device D *ibmq_boeblingen*. The results for the other IBM quantum devices A, B, and C, are shown in Fig. 22 (Appendix D). For the IBM quantum device D (*ibmq_boeblingen*), the values are $a = 6.2774 \pm 0.01502$ and $b = 1.5086 \times 10^{-9} \pm 0.6104 \times 10^{-9}$ s. How we arrived at this dependence is discussed later in Sec. V. Using this data-driven model allows us to use reduce the number of parameters in the estimation and hence reduce the associated computational cost of the optimization. In the following sections, we will denote this model by $\Delta\hat{t}_{\text{eff}}(\boldsymbol{\theta})$.

3. Decoherence

Recalling our discussion in Sec. II B, we model decoherence as a depolarization channel acting on the quantum state $\rho(t) = \exp(-iHt)\rho(0)$ produced as a result of Hamiltonian evolution for a time duration t. The resulting state [from Eq. (11)] was given by

$$\mathcal{E}(\rho(t)) = (1 - p_d(t))\rho(t) + p_d(t)\frac{I}{2^n}. \tag{47}$$

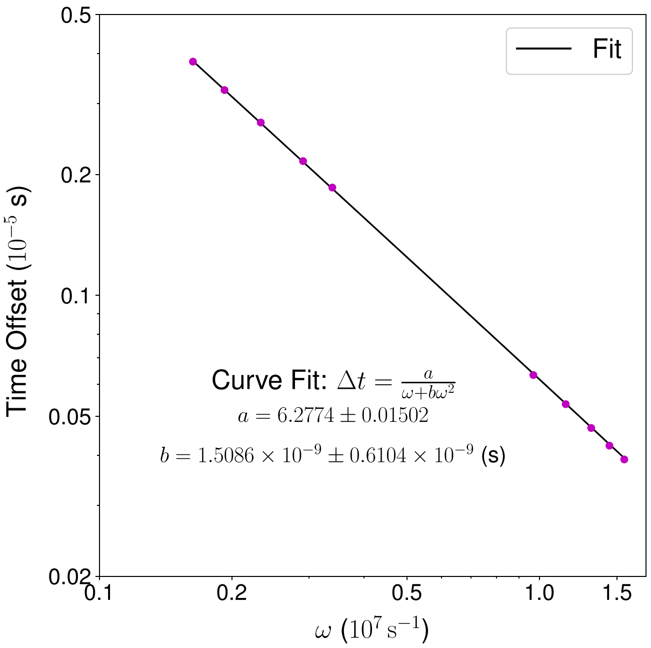

FIG. 6. Dependence of the time offset Δt on parameters ω for IBM quantum device D *ibmq_boeblingen*. The plotted data points correspond to driving the device under different conditions and hence different cross-resonance Hamiltonians. The imperfect pulse shaping model extracted from these experimental data points is shown by a fit and this is later used in the MLE.

One approach to obtain a description of $p_d(t)$ is to assume the functional form $1 - \exp(-(t-t_0)/\mu)$, resulting from a Poisson process with rate μ as described in Sec. II B and then estimate μ from the training examples after incorporating this into the MLE. Here, we describe a model for $p_d(t)$ using the measured device properties of T_1 and T_2 times of each qubit.

We consider an independent noise model on each of the two qubits used for implementing a cross-resonance gate. Let us denote the amplitude damping and phase damping of kth qubit as $\mathcal{E}_{a,k}$ and $\mathcal{E}_{p,k}$ respectively. They have the following Kraus operators:

$$\begin{aligned}\mathcal{E}_{a,k} &: \left\{\begin{bmatrix}1 & 0\\ 0 & \sqrt{1-\gamma_{a,k}}\end{bmatrix}, \begin{bmatrix}0 & \sqrt{\gamma_{a,k}}\\ 0 & 0\end{bmatrix}\right\},\\ \mathcal{E}_{p,k} &: \left\{\begin{bmatrix}1 & 0\\ 0 & \sqrt{1-\gamma_{p,k}}\end{bmatrix}, \begin{bmatrix}0 & 0\\ 0 & \sqrt{\gamma_{p,k}}\end{bmatrix}\right\}\end{aligned} \tag{48}$$

where

$$\gamma_{a,k} = 1 - \exp(-\Gamma_{a,k}t), \quad \gamma_{p,k} = 1 - \exp(-\Gamma_{p,k}t) \tag{49}$$

with

$$\Gamma_{a,k} := \frac{1}{2\mathrm{T}_{1,k}}, \quad \Gamma_{p,k} := \frac{1}{\mathrm{T}_{\phi,k}} \tag{50}$$

where $\mathrm{T}_{1,k}$ is the T_1 time of the kth qubit and $\mathrm{T}_{\phi,k}$ is the pure dephasing rate of the kth qubit related to the T_2 time of the kth qubit as

$$\frac{1}{\mathrm{T}_{\phi,k}} = \frac{1}{\mathrm{T}_{2,k}} - \frac{1}{2\mathrm{T}_{1,k}}. \tag{51}$$

TABLE II. Comparison of different decoherence models in fitting Rabi oscillations inferred from experimental data collected from IBM quantum Device D. Kullback-Leibler divergence (KL divergence) is computed as $D_{\mathrm{KL}}(p_{\mathrm{data}}||p_{\mathrm{model}})$ where p_{data} is the probability inferred from data and p_{model} is that predicted from the model.

Decoherence model	Root mean squared error	KL divergence
No decoherence	0.109321	0.012370
Single-qubit decoherence model	0.095903	0.007408
Single-parameter model	0.091644	0.006515
Two-qubit decoherence model	0.090953	0.006392
Two-parameter model	0.084289	0.005376

The overall noise operator acting on the two qubits is then given by [77]

$$\mathcal{E} = \otimes_{k=1}^{2}(\mathcal{E}_{a,k} \circ \mathcal{E}_{p,k}) \tag{52}$$

where $\circ$ indicates taking a composition of the two noise operators. The probability $p_d(t)$ can be based on the unitarity [78] of the noise operator $\mathcal{E}$, which is a completely positive linear map quantifying the coherence of the noise operator. The probability $p_d(t)$ is then given by

$$\begin{aligned}1 - p_d(t) = 1 + \frac{1}{15}\big(&\gamma_{a,1}^2(3\gamma_{a,2}(\gamma_{p,2}-2) - 4\gamma_{p,2} + 7)\\ &+ 4\gamma_{a,1}(\gamma_{p,1}-2)\big(\gamma_{a,2}^2 + \gamma_{a,2}(\gamma_{p,2}-2)\\ &- \gamma_{p,2} + 2\big) + \gamma_{a,2}^2(7 - 4\gamma_{p,1})\\ &- 4\gamma_{a,2}(\gamma_{p,1}-2)(\gamma_{p,2}-2) + 4\gamma_{p,1}\gamma_{p,2}\\ &- 8\gamma_{p,1} - 8\gamma_{p,2}\big).\end{aligned} \tag{53}$$

This can be further generalized to n-qubit system (see Appendix G of [77] for details). Let the measurement of Hamiltonian evolution for time t followed by this noisy depolarization channel be $\tilde{y}$ and the corresponding Rabi oscillation $\tilde{p}_{\mathrm{rabi}}$. Note that the Rabi oscillations in this case are related to the noiseless case [from Eq. (42)] as

$$\tilde{p}_{\mathrm{rabi}} = (1 - p_d(t))p_{\mathrm{rabi}}. \tag{54}$$

In Table II, we give the RMSE between Rabi oscillations obtained using different decoherence models assuming we know the true Hamiltonian parameters and Rabi oscillations inferred from data. For the *single parameter model*, we consider a prefactor of $(1 - p_d(t)) = \exp(-(t-t_0)/\mu)$ with the single parameter μ on the Rabi oscillations obtained using no decoherence model. This parameter μ is then estimated from the data and is found to be $(7.75 \pm 0.91) \times 10^{-5}$ s. In the *two-parameter model*, this is allowed to vary with the state preparation operator U being applied. For $U = \sigma_I \otimes \sigma_I$, we have $\mu = (5.52 \pm 0.82) \times 10^{-5}$ s and for $U = \sigma_X \otimes \sigma_I$, $\mu = (2.51 \pm 0.59) \times 10^{-5}$ s. It is advantageous to use such models due to the low number of parameters present and when the T_1 or T_2 times of the different qubits are not available. However, if these times are available, one can use the two-qubit decoherence model or the two parameter model as we note from the values of RMSE in Table II.

D. Estimation procedures for learning CR Hamiltonians

We have so far described the likelihood of different measurement outcomes given different queries (Sec. IV B 2) to the two-qubit cross-resonance Hamiltonian [Eq. (30)], and the different noise sources or nonidealities on the IBM quantum devices. We now describe our estimation procedures for estimators described earlier in Sec. II A 4, which utilizes this likelihood function for estimating the CR Hamiltonian parameters from the training data $D = (X, Y)$ generated during learning. While all the estimators described here can be adopted with HAL algorithms (Sec. III), it should be noted that the best performance is expected with the maximum-likelihood estimator (MLE) due to the query criteria used in HAL-FI/HAL-FIR.

1. Regression

We perform regression on Rabi oscillations [Eq. (42)] (which is just the difference in likelihood for $y = 0$ and $y = 1$) inferred from the measurement outcomes. Let us consider the parametrization of $\mathbf{\Lambda}$. This is more useful to work with as the frequencies of oscillation in e^{-iHt} are determined by ω_0 and ω_1 unlike $\mathbf{J}$ where it is determined by all the parameters. Instead of directly performing regression as described in Sec. II A 4 b, we divide our estimation procedure into multiple steps. We create initial estimates of $\omega_{0,1}$ independent of the other parameters through fast Fourier transform (FFT) before performing a full regression.

We compute the Rabi oscillations $\hat{p}_{\text{rabi}}$ for each query x through the corresponding measurement outcomes by solving the optimization problem of Eq. (39). Initial estimates of the parameters $\omega_{0,1}$ are then obtained by applying a discrete (fast) Fourier transform to the Rabi oscillations. These initial estimates are then refined by fitting regression equations of the form $A\cos(\omega t) + B\sin(\omega t) + C$ to the Rabi oscillations, where the fit minimizes the total L_2 error, the coefficients A, B, and C for each Rabi oscillation are estimated using linear least-squares regression, and a bracketed gradient-based search is performed to refine the estimates of $\omega_{0,1}$. The corresponding optimization problem can be framed as minimizing the following residual error:

$$\min E(\mathbf{A}, \omega_{0,1}) = \sum_{t\in\mathcal{T}} (p_{\text{rabi}}(t) - \mathbf{A}\mathbf{\Omega}(\omega t))^2, \tag{55}$$

where the coefficients $\mathbf{A} = (A, B, C)$ are known functions of $\delta_{0,1}, \phi_{0,1}$ through the analytical forms of the Rabi oscillations for the query space considered [see Eq. (42)] and $\Omega(\omega t)$ is a vector of cosines and sines (fully described in Appendix C). Thus, we can then obtain estimates for $\delta_{0,1}, \phi_{0,1}$ from the A, B, and C coefficients of the regression equations for each of the Rabi oscillations. Finally, we obtain an estimate $\hat{\mathbf{\Lambda}}$ by fixing the values of $\omega_{0,1}$ and carrying out a gradient descent procedure using the same cost function. If desired, we output the Hamiltonian parameter estimate $\hat{\mathbf{J}}$ by transforming $\hat{\mathbf{\Lambda}}$ appropriately [Eq. (32)].

2. Maximum-likelihood estimation

The MLE parameter estimate $\hat{\boldsymbol{\theta}}$ obtained through solving Eq. (5) for the CR Hamiltonian incorporating the Hamiltonian model description and presence of different noise sources is

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -\frac{1}{N}\sum_{k=1}^{N}\log\left[\sum_{y\in\{0,1\}}\left(p_{\tilde{\text{y}}|\text{y}}(\tilde{y}^{(k)}|y)\sum_{z\in\{0,1\}}[(1-p_d(t^{(k)}))|\langle yz|M^{(k)}e^{-iH(\boldsymbol{\theta})(t^{(k)}+\Delta\hat{t}_{\text{eff}}(\boldsymbol{\theta}))}U^{(k)}|00\rangle|^2 + \frac{1}{4}p_d(t^{(k)})]\right)\right], \tag{56}$$

or

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -\frac{1}{N}\sum_{k=1}^{N}\log\left[\sum_{y\in\{0,1\}}\left(p_{\text{c}|\text{y}}(c^{(k)}|y)\sum_{z\in\{0,1\}}[(1-p_d(t^{(k)}))|\langle yz|M^{(k)}e^{-iH(\boldsymbol{\theta})(t^{(k)}+\Delta\hat{t}_{\text{eff}}(\boldsymbol{\theta}))}U^{(k)}|00\rangle|^2 + \frac{1}{4}p_d(t^{(k)})]\right)\right], \tag{57}$$

depending on how the readout noise is modeled. The imperfect pulse-shaping model $\Delta\hat{t}_{\text{eff}}(\boldsymbol{\theta})$ was discussed in Sec. IV C 2 and $p_d(t)$ is the probability of depolarization associated with the two-qubit decoherence model (Sec. IV C 3). The choice of the MLE [Eq. (56) or (57)] for each IBM quantum device will be specified in Sec. V.

The MLE problem [Eqs. (56) and (57)] in either of the two different parametrizations of $\mathbf{J}$ and $\mathbf{\Lambda}$ is nonlinear and nonconvex. An example of the energy landscape of the log-likelihood loss function for the IBM quantum device *ibmq_boeblingen* is shown in Fig. 21 of Appendix C. The presence of multiple local minima makes the MLE problem in general challenging to solve. To ensure that we converge to the global minimum and do not get stuck in a local minimum during estimation, we divide the estimation into multiple stages. In the first stage, we obtain an initial estimate, which is refined over subsequent stages. The estimation procedure is summarized in Algorithm 4.

During learning, this initial estimate may be available from the previous round. When such an initial estimate is not available, we use the estimation based on regression as discussed earlier in Sec. IV D 1, and summarized in lines 1–6 of Algorithm 4. This initial estimate is then used as an initial condition to the MLE solve [Eq. (56)]. In Algorithm 4, we have denoted the negative log-likelihood loss function, which appears in the optimization problem of Eq. (56) as L. We solve the MLE problem using the optimizer of stochastic gradient method of ADAM [79], which encourages getting out of any local minima. The parameter estimate produced by ADAM is then refined using the second-order quasi-Newton method of L-BFGS-B [80]. The computational complexity of our MLE estimation procedure is dominated by ADAM. This motivates us to directly use L-BFGS-B for estimation for latter batches during learning. The full description, computational details and extensions of the estimation procedure is given in Appendix C.

Algorithm 4. MLE estimation procedure for Hamiltonian learning

```
Input: Training examples of size $m$: $D = \{(X, Y)\} = \{(x_i, y_i)\}_{i\in[m]}$, initial condition $\hat{\mathbf{J}}_0$ (optional)
Output: $\hat{\mathbf{J}}$
if no input of $\hat{\mathbf{J}}_0$ then        ▷ get an initial estimate through regression
    Obtain Rabi oscillations $p_{\mathrm{rabi}}$ from $D$ by solving Eq. (39)
    $\omega_{0,1} \leftarrow \mathrm{FFT}(p_{\mathrm{rabi}})$        ▷ FFT of Rabi oscillations $p_{\mathrm{rabi}}$ over $\mathcal{T}$, see Appendix C for details
    $(\delta_{0,1}, \phi_{0,1}) \leftarrow p_{\mathrm{rabi}}(\Omega(\omega_{0,1}t))^{-1}$        ▷ Regression on Rabi oscillations given $\omega_{0,1}$
    Refine estimate of $\hat{\mathbf{\Lambda}}_0$ through gradient descent on $E(\mathbf{A}, \omega_{0,1})$ in Eq. (55)
    Get $\hat{\mathbf{J}}_0$ by transforming $\hat{\mathbf{\Lambda}}_0$ [Eq. (32)]
$\hat{\mathbf{J}} \leftarrow \arg\min_{\mathbf{J}} L(\mathbf{J}; X, Y)$        ▷ using $\hat{\mathbf{J}}_0$ as a guess
return $\hat{\mathbf{J}}$
```

3. Bayesian estimation

For Bayesian estimation, we use a particle filtering algorithm [42], also called sequential Monte Carlo (SMC) algorithm. In SMC, a distribution [say $p(r)$] is discretely approximated with a distribution that has support only over a finite number of points (say n_p) called particles,

$$p(r) = \sum_{k=1}^{n_p} w_k \delta(r - r_k), \tag{58}$$

where w_k is the weight of the kth particle, and r_k is the location of the kth particle. Note that the vector of weights (w_k) can also be thought of as a vector of probabilities, satisfying $0 \leqslant w_k \leqslant 1$ and $\sum_{k=1}^{n_p} w_k = 1$. An particle filter is then specified by weights and locations $\{(w_k, r_k)\}_{k\in[n_p]}$ over the set of n_p particles. This can be used to compute the expectation value of a function g as $\sum_{k=1}^{n_p} w_k g(r_k)$.

For learning the CR Hamiltonian, we will be interested in tracking the distribution $p(\mathbf{J}|D)$ of the Hamiltonian parameters given the training data D being collected. It is then beneficial to think of the particle filter as $\{(w_k, \mathbf{J}_k)\}_{k\in[n_p]}$. Considering the prior distribution as $p(\mathbf{J})$, the initial weights of the particles can be set as $w_k = 1/n_p$ and the locations as random samples $\mathbf{J}_k$, sampled from this prior. With this prior distribution, we can then carry out Bayes rule [Eq. (7)] to compute the posterior distribution $p(\mathbf{J}|D)$ over the Hamiltonian parameters $\mathbf{J}$ described by particles $\{(w'_k, \mathbf{J}'_k)\}_{k\in[n_p]}$ by setting $\mathbf{J}'_k = \mathbf{J}_k$ and

$$w'_k = \frac{w_k p(Y|X, \mathbf{J}_k)}{\sum_k p(Y|X, \mathbf{J}_k)}. \tag{59}$$

In practice, iteratively carrying out Baye's rule as above leads to numerical instabilities due to limited resolution and weights shrinking to zero. Increasing the number of particles n_p obviously delays this while improving the accuracy of the evaluation of expectations of functions using the particles and posterior distributions in the SMC algorithm. Stability is ensured by resampling techniques, which adaptively changes locations $\mathbf{J}_k$ of particles to higher weight regions. In our implementation, we use the Liu-West algorithm [81].

Finally, we can output a point estimate from the posterior particles

$$\hat{\mathbf{J}} = \sum_{k=1}^{n_p} w'_k \mathbf{J}'_k, \tag{60}$$

which is the mean of the posterior distribution (Sec. II A 4 c) and minimizes the RMSE.

We use the implementation of SMC in the Qinfer package [38] for our purposes. We thus refer the reader to [37,38] for details on the SMC algorithm used for Hamiltonian learning. We specify the prior distribution $p(\mathbf{J})$ and the number of particles n_p chosen for different IBM quantum devices and learners in Sec. V.

E. Implementation of HAL algorithm for learning CR Hamiltonians

We now discuss the details of the HAL-FI algorithm implemented for learning the CR Hamiltonian on the IBM quantum devices. As the implementation of HAL-FIR for this application is very similar, it is omitted. Moreover, we will consider the MLE estimator here but the other estimators (Sec. IV D) can also be adopted.

In our implementation of the HAL-FI algorithm in experiments on learning the CR Hamiltonian, we consider the following inputs. The initial query space, which may be changed during the course of the HAl-FI algorithm during training is $\mathcal{Q}^{(0)} = \mathcal{M} \times \mathcal{U} \times \mathcal{T}^{(0)}$ with $\mathcal{M}$ and $\mathcal{U}$ as described in Sec. IV B 2. Here, we explicitly denote the superscript on $\mathcal{T}$, which is initially set to $\mathcal{T}^{(0)}$ but may change during training if an adaptive query space strategy is employed. We set $\mathcal{T}^{(0)}$ to be the 81 equispaced times in the interval $[10^{-7}, 6 \times 10^{-7}]$ s. We consider the initial number of queries as $N_{\mathrm{tot}}^{(0)} = 2430$ (or five times the number of different queries in the initial query space $\mathcal{Q}^{(0)}$), a constant batch size $N_b = 486$ (or the number of queries in the initial query space $\mathcal{Q}^{(0)}$), and the initial query distribution $q^{(0)}$ as the uniform random distribution over $\mathcal{Q}^{(0)}$.

The initial set of training examples $(X^{(0)}, Y^{(0)})$ are obtained by sampling $X^{(0)}$ from $\mathcal{Q}^{(0)}$ with respect to $q^{(0)}$ in Line 1 of Algorithm 1 and collecting the corresponding set of measurement outcomes $Y^{(0)}$ through queries to the CR Hamiltonian on the IBM quantum device in Line 2. The set of training examples $(X^{(i)}, Y^{(i)})$ is progressively increased (Line 10) during learning by adding N_b queries $X_q^{(i)}$ sampled from query distribution $q^{(i)}$ chosen by HAL-FI in Line 6 and collecting the corresponding measurement outcomes $Y_q^{(i)}$ in Line 9. The learning is continued until our query budget is expended or the desired learning error is achieved.

To determine the initial parameter estimate $\hat{\boldsymbol{\theta}}^{(0)}$ from the initial set of training examples (Line 4) and subsequent $\hat{\boldsymbol{\theta}}^{(i)}$ from $(X^{(i)}, Y^{(i)})$ (Line 11), we solve the MLE [Eq. (5)] for the CR gate incorporating the Hamiltonian model description and presence of different noise sources.

The parameter estimates $\hat{\boldsymbol{\theta}}$ obtained after solving the MLE problem are used by HAL-FI to construct the Fisher information matrices based on the model and obtain the query distribution $q^{(i)}$ by solving the SDP program of Eq. (23) in Line 6. The expressions for the Fisher information

TABLE III. Summary of different learning scenarios.

Learning scenario	Query space	Query distribution
Passive learning	Fixed	uniformly random
Active learning with fixed query space	Fixed	q through Eq. (19)
Active learning with adaptive query space I	Linearly growing $\mathcal{T}$	q through Eq. (19)
Active learning with adaptive query space II	Exponentially growing $\mathcal{T}$	q through Eq. (19)

matrices for different queries (Sec. IV B 2) considering the CR Hamiltonian and noise models are given in Appendix A. The query space $\mathcal{Q}^{(i)}$ used in Lines 6 and 7 depend on the querying strategy and hence the learning scenario we consider.

As described in Sec. III A, we can define four different learning scenarios based on the presence of an active learner and how the query space is adaptively changed during learning. In passive learning, an active learner is not present and we set the query distribution to the uniformly random distribution over $\mathcal{Q}$. In the case of active learning with fixed query space, the query space remains fixed during training, i.e., $\mathcal{Q}^{(i)} = \mathcal{Q} \forall i$, and the query distribution is determined by solving Eq. (23) using the current estimate of the parameters $\hat{\boldsymbol{\theta}}$ over this fixed query space. When considering active learning with an adaptively growing query space, we consider two different situations on the basis of how the query space is changed between batches during learning. We consider two cases: (i) $\mathcal{Q}^{(i)}$ grows linearly by linearly increasing the $\mathcal{T}^{(i)}$ between batches and (ii) $\mathcal{Q}^{(i)}$ grows exponentially by doubling the allowed set of system interaction time $\mathcal{T}^{(i)}$. The query distribution is then determined by solving the corresponding SDP problem of Eq. (23) over the query space $\mathcal{Q}^{(i)}$ corresponding to the ith batch. These different learning scenarios for HAL-FI are summarized in Table III.

V. RESULTS

We now present the results of the experiments described in Sec. IV and show that they support the claims made in Sec. II A. In this section, we compare the active learner HAL introduced in Sec. III against the passive learner (Sec. II C 1 d) equipped with different estimators (Sec. IV D) and the sequential active learner of Qinfer (Sec. II C 5) in learning the CR Hamiltonian through experiments on different IBM quantum devices. We finally analyze the results of these experiments to comment on the scaling behavior, classical computational cost and query advantage of the different learners.

In Sec. V A, we first describe the datasets used for Hamiltonian learning, hyperparameters of the estimators (Sec. IV D) used in experiments, and the experimental protocol for evaluating the performance of the learners. In Sec. V B, we describe the implementation of the sequential active learner of Qinfer, which is also used as a baseline for assessing the performance of HAL-FI. In Sec. V C, we show results of the learners on these different datasets. In Sec. V D, we describe how HAL-FI can be used to achieve Heisenberg (or super-Heisenberg) limited scaling and evaluate the query advantage of HAL-FI over the baseline considering different learning scenarios.

A. Data and experiment protocol

In this section, we describe the different kinds of datasets that were used in assessing the performance of the HAL-FI algorithm and how they were collected. We then summarize the parameters of the cross-resonance Hamiltonian and the noise sources discussed in Sec. IV C for the different IBM quantum devices (Sec. IV B 1).

1. Datasets from IBM quantum devices

The different datasets that we use for Hamiltonian learning are a combination of experimental data collected from the IBM quantum devices described in Sec. IV B 1 and that collected from a *simulator*, which we will describe in Sec. V A 3.

Experimental data is collected from the different IBM quantum devices according to the query space described in Sec. IV B 2. The set of evolution times $\mathcal{T}$ is set to 81 equispaced times in the interval $\mathcal{T} = [10^{-7}, 6 \times 10^{-7}]$ s. For IBM quantum devices A, B, and C, there are 200 measurement outcomes (or shots) for each query $\mathbf{x} \in \mathcal{Q}$. For IBM quantum device D *ibmq_boeblingen*, there are 512 measurement outcomes for each query. Recall from our discussion of the Hamiltonian learning framework in Sec. II A and HAL algorithm in Sec. III B, the outputs of our queries are not expectation values but rather single shot readouts of the target qubit.

The experimental data is then collected and made available as an offline dataset that the active learner can query. Unlike deploying an active learner in real-time where a particular query can be made to the system unlimited number of times, using experimental datasets imposes the additional constraint of the number of times a query can be made by the active learner due to the limitation on the number of measurement outcomes available for each query. In Appendix B, we discuss how we handle this constraint during query optimization.

2. Parameters of the CR Hamiltonian and noise sources for IBM quantum devices

Considering the entire collected experimental datasets for each IBM quantum device (Sec. IV B 1) as training data, we compute the Hamiltonian parameters $\mathbf{J}/\mathbf{\Lambda}$, and that of the different noise sources using the estimation procedure specified in Sec. IV D for the MLE. We solve the MLE of Eq. (56) for IBM quantum device D, which has very low readout noise and use MLE of Eq. (57) for the other IBM quantum devices. A summary of the estimated parameters of the CR Hamiltonian and noise sources on IBM quantum device D *ibmq_boeblingen* is shown in Table IV under different drive configurations. We summarize the estimated parameters for the other devices in Appendix D.

TABLE IV. Summary of estimated CR Hamiltonian parameters for the IBM quantum device D *ibmq_boeblingen* with different drive configurations (Config.) corresponding to amplitude (Amp.) of CR pulse. We give the Hamiltonian parameters in the parametrization $\mathbf{J}$ and the physically relevant frequency components in $\boldsymbol{\Lambda}$. The readout noise is defined by the parameters of r_0 and r_1, which are the conditional probabilities of bit flip given the measurement outcomes are $y = 0$ and $y = 1$ respectively (see Sec. IV C 1).

Drive Config.	CR Amp. (arb. units)	Hamiltonian parameters [$\times 10^6 s^{-1}$]		Noise sources	
		$\mathbf{J} = (J_{IX}, J_{IY}, J_{IZ}, J_{ZX}, J_{ZY}, J_{ZZ})$	(ω_0, ω_1)	Readout (r_0, r_1)	Time Offset $(\Delta t_{\text{eff},0}, \Delta t_{\text{eff},1})$ [ns]
1	0.24	(−3.88, −1.08, −0.24, 5.44, 1.07, 0.21)	(1.57, 9.58)	(0.012, 0.025)	(1965, 289)
2	0.30	(−4.57, −1.47, −0.29, 6.50, 1.39, 0.41)	(1.94, 11.45)	(0.0078, 0.033)	(1581, 226)
3	0.36	(−5.12, −1.65, −0.23, 7.52, 1.66, 0.33)	(2.40, 13.07)	(0.0078, 0.035)	(1267, 203)
4	0.42	(−5.42, −1.95, 0.37, 8.38, 1.90, 0.07)	(2.97, 14.33)	(0.0078, 0.039)	(1016, 182)
5	0.48	(−5.72, −2.13, 0.03, 9.20, 2.15, 0.11)	(3.48, 15.51)	(0.0078, 0.023)	(862, 166)

3. Datasets from simulation

The experimental datasets in Sec. V A 1 contain noise sources other than those modeled even if negligible and may not span a long enough time range over $\mathcal{T}$ for testing different learning scenarios. In order to understand the behavior of the HAL-FI and HAL-FIR algorithms considering all the noise sources are known, we set up a simulator. The advantage of using a simulator over experimental data is that it allows us to assess the limits of the performance of the HAL-FI algorithm in the presence or absence of different noise sources such as decoherence. Studies carried out on the simulator also allow us to test the robustness of the active learner in the presence or absence of different noise sources.

The simulator imitates the different quantum devices but where all the different noise sources are known and perfectly modeled, and which we can query. The Hamiltonian of the simulator is set to that learned from the full set of training examples contained in an experimental dataset collected from a particular quantum device. Thus, we can have simulators for each of the IBM quantum devices A, B, C, and D, under different drive configurations. All the modeled noise sources of readout noise, imperfect pulse-shaping, and decoherence as described in Sec. IV C are included. We apply HAL-FI in real time on the simulator as there is no limitation on the number of times we select a particular query $\mathbf{x} \in \mathcal{Q}$.

In Fig. 7, we compare Rabi oscillations computed from a subset of the experimental data collected from IBM quantum device D *ibmq_boeblingen* under drive configuration 2, and training examples generated on the corresponding simulator. A set of 46 800 training examples are generated from both the experimental data and simulator assuming a uniform query distribution over the query space. These training examples are then used to learn Hamiltonian parameters in each case. The small difference in the predicted model Rabi oscillations, computed using the analytical expressions of Eq. (42) and the Hamiltonian parameters learned from these two different datasets, further indicates that the simulator faithfully represents the collected experimental data.

4. Hyperparameters of estimators for learning CR Hamiltonians

Among the different estimators (Sec. IV D) we use for Hamiltonian learning, some of them require the specification of some hyperparameters. In our estimation procedure for MLE (Sec. IV D 2), we set the learning rate for the ADAM solve as 10^{-3} and use only the quasi-Newton method after 50 batches. In the Bayesian SMC method (Sec. II A 4 c), we use n_p particles to track the distribution over $p(\tilde{\mathbf{J}}|D)$ where $\tilde{\mathbf{J}}$ is the normalized version of the Hamiltonian parameter vector $\mathbf{J}$ by 10^6. We choose this normalization factor as this is the order of magnitude expected for these parameters under these drive conditions [73]. We then set the prior distribution over the parameters as $p(\tilde{\mathbf{J}})$ as an uniform distribution over $[-10, 10]^6$. On the simulator, we set $n_p = 10^4$ and on the experimental data, we set $n_p = 10^5$. These values were obtained

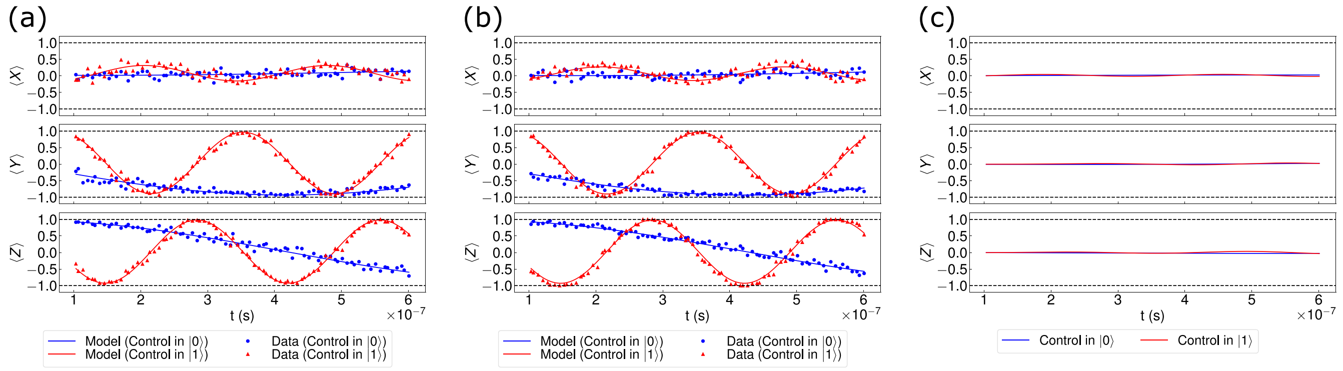

FIG. 7. Comparison of Rabi oscillations computed from data (46 800 queries) of IBM quantum device D *ibmq_boeblingen* from the (a) collected experimental data, and (b) simulator. In (c), we plot the difference in the model Rabi oscillations predicted from the Hamiltonians learned in (a) and (b). In the subplots of (a)–(c), we plot Rabi oscillations (or difference) for different measurement operators M (rows), preparation operators U (colors), and evolution times t (x axis), corresponding to the query space described in Sec. IV B 2.

by increasing the number of particles on each oracle until we saw the expected convergence behavior with a passive learner equipped with the SMC method as an estimator.

5. Protocol for comparing performance of Hamiltonian learning methods

Query complexity is used for comparing the performance of different learners on the simulator or oracle with access to an experimental dataset. In particular, our main goal is to extract the scaling of the query complexity with respect to the root mean square error (RMSE) of Hamiltonian parameters $\mathbf{J}$. On the simulator as we have access to the truth $\mathbf{J}^\star$, we can compute the RMSE directly. On the experimental data, we use the empirical RMSE, computed at each round of learning as

$$\begin{aligned} \text{RMSE} &= \sum_{i=1}^{m} \mathbb{E}\left[\left(\frac{\hat{\theta}_i}{\xi_i} - \frac{\theta_i^\star}{\xi_i}\right)^2\right]^{1/2} \\ &\approx \frac{1}{N_{\text{runs}}} \sum_{i=1}^{m} \sum_{k=1}^{N_{\text{runs}}} \left[\left(\frac{\hat{\theta}_i^{(k)}}{\xi_i} - \frac{\theta_i^\star}{\xi_i}\right)^2\right]^{1/2}, \end{aligned} \qquad (61)$$

where we approximate the expectation by an average over parameter estimates from N_{runs} runs and the true parameter values $\boldsymbol{\theta}^\star$ by the mean of these runs. The normalization factors ξ_i are selected to be $10^6\ \text{s}^{-1}$ for all i as the components of $\mathbf{J}$ with highest magnitude are expected to the order of $10^{6\pm1}$ [73] for these IBM quantum devices under these drive configurations. We implement the following experimental protocol. For each of the quantum devices described in Sec. IV B 1, we compute the empirical RMSE for the learners from 200 independent runs of the simulator and 500 independent runs on the experimental dataset. These number of runs on each oracle were required to obtain accurate scalings of trends in RMSE with number of queries and ensure the uncertainty (or two standard deviations) of each scaling was at most 10%. In each run, we carry out the Hamiltonian learning algorithm for the different learning scenarios as detailed in Sec. IV E and summarized in Table III. Additionally, we track the testing error of the learner with number of queries N. The so obtained trends are used to comment on the robustness of the estimation procedure used for MLE (Sec. IV D) and the benefits of using the active learner HAL-FIR for making predictions of queries to the Hamiltonian (Problem II A 3 b) over a baseline. The testing error is computed empirically as well on a testing dataset collected from the simulator or experimental dataset using p_{test}.

B. Sequential active learner: Qinfer

In this section, we describe the sequential active learner of Qinfer [37,38] that was earlier introduced in Sec. II C 5 and how it is used in our numerical experiments on learning the CR Hamiltonian.

The sequential active learner in [37,38], which we refer to simply as Qinfer [82] uses Bayes risk [Eq. (26)] as a query criteria. In our implementation, the Bayes risk is computed by running a *hypothetical update* over a set of *risk* particles in SMC (Sec. IV D 3) for each query in the query space $\mathcal{Q}$. These risk particles are a separate set of particles than those used to track the distribution $p(\boldsymbol{\theta}|D)$ in SMC. The weights and locations of these risk particles are set to the same values as those of the particles partaking in estimation, however, before computation of the Bayes risk over all queries and after every estimation step. Further details of Bayes risk computation can be found in [38].

For a fair and systematic comparison of Qinfer against HAL-FI on the simulator and experimental data, we run Qinfer sequentially over batches of queries. Running Qinfer sequentially over $\sim 10^5$ shots/queries would be computationally very expensive, as we would need to update the particles in SMC after collecting just one measurement outcome and then compute Bayes risk over all the queries in $\mathcal{Q}$. Instead, we issue a batch of N_b queries, all of which are the same query and the one with the lowest Bayes risk. The corresponding measurement outcomes are then used to update the particles in SMC before computing Bayes risk using the updated risk particles and issuing another batch of queries.

On experimental data, we also need to ensure that query constraints are respected due to limited number of measurement outcomes available for each query. We do this by breaking down a batch of queries into mini-batches if enough number of measurement outcomes is not available for the request query with the lowest Bayes risk. Further details are given in Appendix B.

Finally in Qinfer, we use the same prior distribution and same number of particles as in the case of the passive learner equipped with the Bayesian SMC estimator (Sec. IV D 3). On the simulator, we set $n_p = 10^4$ and on the experimental data we set $n_p = 10^5$.

C. Performance of Hamiltonian learning methods

In this section, we assess the performance of the different Hamiltonian learning methods introduced so far using the protocol described in Sec. V A 5 in tackling the learning problems posed in Sec. II A. We report results of the different Hamiltonian learning scenarios (Table III): (i) passive learner with estimation based on FFT and linear regression, (ii) passive learner with an estimation procedure to solve the MLE problem, (iii) active learner in fixed query space, and (iv) active learner in an adaptively growing (linearly) query space. Results of the exponentially growing query space are postponed to Sec. V D 1 where we comment on the achievability of Heisenberg limited scaling. We use the HAL-FI algorithm, which was discussed in Sec. II C in the latter active learning approaches. We also show results of learners with different estimations based on FFT and linear regression (Lin. Reg.), maximum-likelihood (MLE), and Bayesian estimation using the sequential Monte Carlo (SMC) method. We additionally compare HAL-FI, which proposes batches of queries during learning against the sequential active learner Qinfer (Sec. V B), which proposes queries one at a time using Bayes risk as a query criteria.

We present the convergence behavior of each algorithm during learning under different regimes. For each scenario, we show trends of learning error (RMSE) with number of queries. These trends indicate the performance of each learning algorithm, culminating in evidence of query advantage. For brevity of presentation, we focus on the results obtained

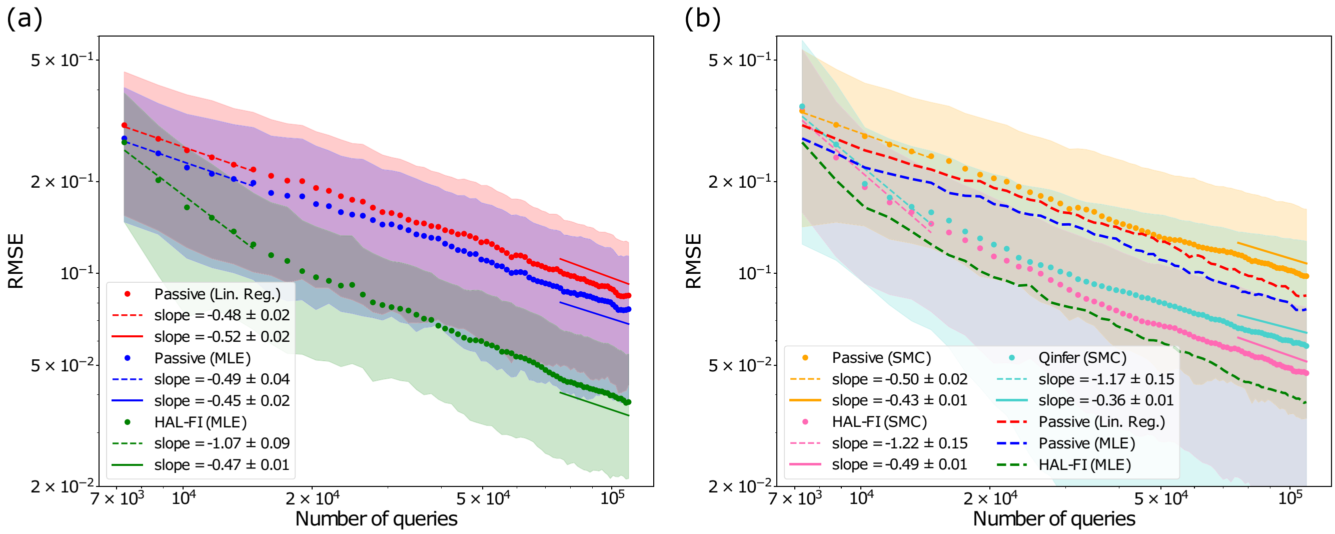

FIG. 8. Scaling of RMSE with number of queries for different learners on a simulator when MFC is satisfied. In (a), we compare the trends in RMSE of HAL-FI against the passive learner with estimations based on MLE and linear regression (Lin. Reg.). In (b), we compare the trends in RMSE of HAL-FI using the Bayesian estimator of sequential Monte Carlo (SMC) against HAL-FI using MLE, Qinfer using Bayes risk as the query criteria and passive learners equipped with different estimators. In (a) and (b), estimators of each learner are indicated in brackets. Slopes indicate the scaling of RMSE with number of queries in the finite sample and asymptotic sample regimes. Filled in areas indicate the respective errors on trends for each learner. In (b), we do not show slopes and errors for learners shown earlier in (a) on the left, and their trends are shown for visual reference.

from IBM quantum Device D (*ibmq_boeblingen*) under the drive configuration 2.

We focus on the Hamiltonian learning task of model inference (II A 3 a) and touch upon prediction against a testing distribution (II A 3 b) to show that HAL is a general framework for tackling both problems when equipped with the appropriate query optimization. We show results on the simulator for two different cases of time range $\mathcal{T}$ of the query space $\mathcal{Q}$. The time range $\mathcal{T}$ should be chosen such that you can estimate the Hamiltonian parameters of interest $\hat{\boldsymbol{\theta}}$. For a given $\mathcal{T}$, the non-zero frequencies that can be successfully detected using FFT [83,84] range from $1/\Delta T$ to $F_s/2 - 1/\Delta T$ at increments of $1/\Delta T$ where ΔT is the length of the time interval $\mathcal{T}$ and F_s is the sampling rate. F_s is decided according to the Nyquist criterion to ensure that aliasing does not occur. Accordingly, ΔT must be sufficiently long to see a single cycle of the sinusoid corresponding to the lowest frequency. We call this as the minimum-frequency criteria (MFC). Note that when using other estimation methods based on MLE or Bayesian estimation methods, $\mathcal{T}$ is not required to satisfy these properties.

We thus show results on the simulator where $\mathcal{T}$ is such that the MFC for frequency estimation (using FFT) is satisfied and when MFC is not necessarily satisfied. Moreover, in Appendix C, we describe how standard FFT can be modified to detect lower frequencies than MFC allows. This is a build-up to our comparison on the experimental dataset where $\mathcal{T}$ is such that MFC is not necessarily satisfied. In the following results, we plot the learning error (RMSE/testing error) versus number of queries made on a log-log scale so that the slope s of the plotted lines can be directly interpreted as the scaling of learning error with complexity $\epsilon \sim N^s$.

(a) Simulator and minimum-frequency criteria is satisfied. We consider the query space as defined in Sec. IV B 2 with $\mathcal{T}$ set to be the 243 equispaced times in the interval of $[10^{-7}, 18 \times 10^{-7}]$ s. A comparison of different learners for Hamiltonian learning considering this query space is shown in Fig. 8. For the passive learners (with any estimator), we observe a scaling of $\epsilon \sim 1/\sqrt{N}$ or $N \sim \epsilon^{-2}$ in RMSE with number of queries. This is in agreement with the SQL. The approximately constant gap between the passive learner when using an estimator based on MLE and the passive learner when using an estimator based on linear regression corresponds to a constant query reduction. This illustrates how a query advantage can be obtained by changing estimation for Hamiltonian learning.

In Fig. 8(a), we observe two different scalings for the HAL-FI algorithm with MLE, an initial scaling, which is higher than SQL and similar to Heisenberg limited scaling, and a scaling of SQL in the asymptotic query regime. This shows that depending on the desired learning error, we can expect to see a higher rate of convergence. However, asymptotically the number of queries N required by the active learner HAL-FI is a constant fraction of that required by the passive learner.

We can also consider HAL-FI using Bayesian estimation for Hamiltonian learning. In Fig. 8(b), we show results comparing learners with the Bayesian estimator SMC, against learners using alternate estimation. Among active learners, we show HAL-FI alongside Qinfer using Bayes risk as a query criteria. Firstly, we note that HAL-FI with SMC outperforms all other learners equipped with the Bayesian estimator including Qinfer. This indicates that HAL-FI can be adopted with different types of estimators in practice, and a query

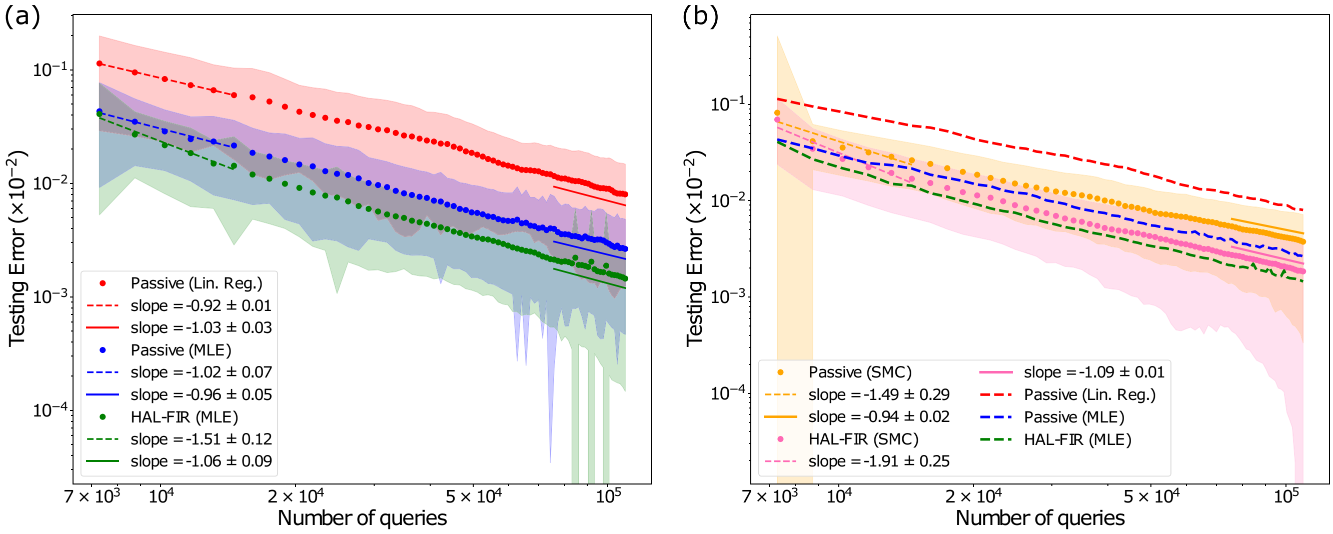

FIG. 9. Scaling of testing error with number of queries for different learners on a simulator when MFC is satisfied. In (a), we compare the trends in testing error of HAL-FIR against the passive learner with estimations based on MLE and linear regression (Lin. Reg.). In (b), we compare the trends in testing error of HAL-FIR using the Bayesian estimator of sequential Monte Carlo (SMC) against HAL-FIR using MLE, and passive learners equipped with different estimators. In (a) and (b), estimators of each learner are indicated in brackets. Slopes indicate the scaling of testing error with number of queries in the finite sample and asymptotic sample regimes. Filled in areas indicate the respective errors on trends for each learner. In (b), we do not show slopes and errors for learners shown earlier in (a) on the left, and their trends are shown for visual reference.

advantage can be obtained. Secondly, similar scaling behavior is seen for the different learners as earlier.

Additionally, we compare the passive learners and active learner HAL-FIR for tackling Problem II A 3 b in Fig. 9. The testing error is computed using Eq. (21) on a testing dataset of 10^5 independent identically distributed (i.i.d.). samples. The testing distribution p_{test} is considered to be known and set to be the uniform distribution over the query space $\mathcal{Q}$. It is assumed that HAL-FIR is given access to this testing distribution. We observe a scaling of $\epsilon \sim 1/N$ for the the passive learners, which is expected as the log-likelihood loss function associated with $\hat{\boldsymbol{\theta}}$ is divergence-free in the asymptotic case [39]. As in the case of HAL-FI for Hamiltonian learning, we observe a higher initial scaling for HAL-FIR and a a scaling of $\epsilon \sim 1/N$ in the asymptotic regime, consistent with that observed for the passive learner.

As discussed in Sec. II A, it is not typical for the testing distribution in Problem II A 3 b to be known and the result here can viewed as a validation of the Hamiltonian model learned and hence the learners on a set of queries sampled using the testing distribution.

We note sudden peaks in uncertainty associated with the passive learner and HAL-FIR using the MLE estimator for higher values of samples. This might be indicative of traveling between multiple local minima in a larger convex hull when solving the MLE [Eq. (56)]. The trends in testing error are not severely impacted by this in expectation indicating that these are rare events and our learners (with their estimation) are robust.

Having made a case for the robustness of the learners and estimation used in this work on the simulator considering a query case $\mathcal{Q}$, which satisfies MFC, we are now in a position to compare the performance of the different learners when the $\mathcal{Q}$ is not guaranteed or known to satisfy such a criteria. In the results that follow, the observations made here are used as a basis for the behavior to expect among the learners.

(b) Simulator and minimum-frequency criteria is not necessarily satisfied. In practice, the range of system evolution times $\mathcal{T}$ corresponding to the query space $\mathcal{Q}$ cannot be known *apriori* to satisfy MFC for frequency estimation using FFT on Rabi oscillation data. It is then crucial for learners equipped with estimation procedures to either: (i) succeed at Hamiltonian learning given this query space or (ii) alert the user that a longer $\mathcal{T}$ is required upon failure to learn a Hamiltonian model. Here, we ensure the former by modifying the standard FFT routine as discussed earlier in Sec. IV D. Further details are also provided in Appendix C. Note that this solve is also carried out as the first step in our estimation procedure for obtaining initial conditions to the MLE solve.

A comparison of the different learners considering a $\mathcal{T}$ that does not satisfy Minimum-Frequency criteria on the simulator is shown in Fig. 10. We set $\mathcal{T}$ to be the set of 81 equispaced times in the interval of $10^{-7}, 6 \times 10^{-7}]$ s. As obtained earlier on the simulator with a longer $\mathcal{T}$, we observe a SQL scaling of $\epsilon \sim 1/\sqrt{N}$ or $(N \sim \epsilon^{-2})$ in RMSE with number of queries for the passive learners combined with estimation based on linear regression, MLE or the Bayesian estimator of SMC. However, there is now a noticeably wider gap in between the trends corresponding to around 80.6% reduction in queries when using the passive learner with the MLE estimator over the passive learner with estimation based on linear regression.

For the HAL-FI algorithm with MLE, we consider cases of when the query space is fixed and when it is adaptively grown by linearly growing the $\mathcal{T}$. For both cases, we see an initial scaling, which is higher than SQL and similar to Heisenberg limited scaling, and a scaling of SQL in the asymptotic query

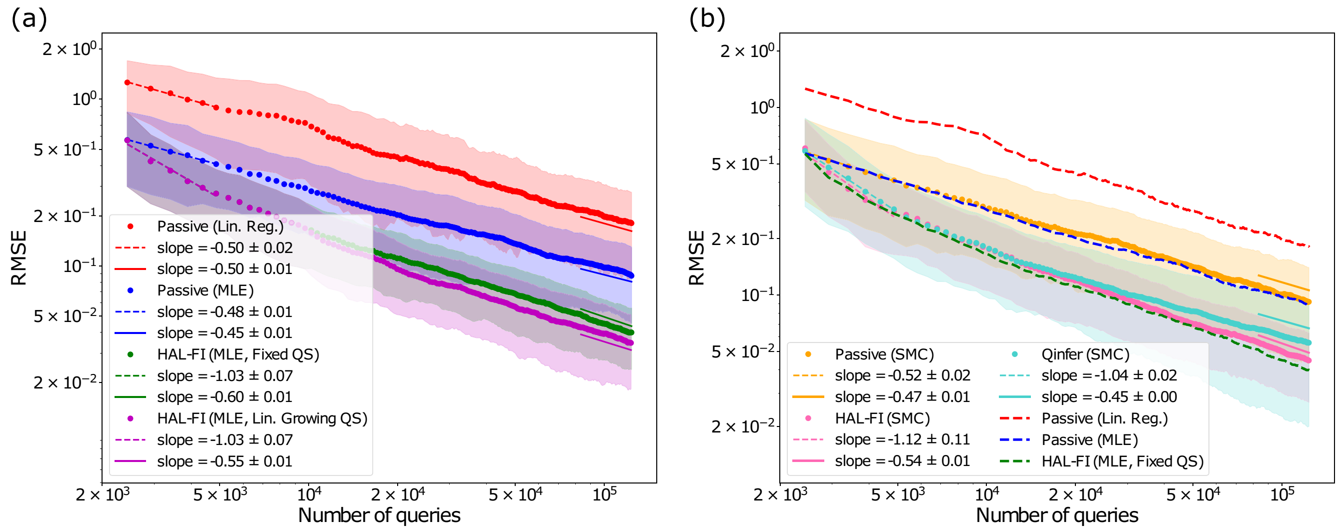

FIG. 10. Scaling of RMSE with number of queries for different learners on a simulator when MFC is not satisfied. In (a), we compare the trends in RMSE of HAL-FI against the passive learner with estimations based on MLE and linear regression (Lin. Reg.). In (b), we compare the trends in RMSE of HAL-FI using the Bayesian estimator of sequential Monte Carlo (SMC) against HAL-FI using MLE, Qinfer using Bayes risk as the query criteria and passive learners equipped with different estimators. In (a) and (b), estimators of each learner are indicated in brackets. Slopes indicate the scaling of RMSE with number of queries in the finite sample and asymptotic sample regimes. Filled in areas indicate the respective errors on trends for each learner. In (b), we do not show slopes and errors for learners shown earlier in (a) on the left, and their trends are shown for visual reference.

regime. This behavior is expected from our observations on the simulator earlier. We note that using HAL-FI combined with a linearly growing query space does not show significant improvement over HAL-FI in the fixed query space. This is due to the low rate at which we adaptively grow the query query space during learning and the fact that growing the query space is only advantageous for learning a subset of the Hamiltonian parameters. We discuss this further in Sec. V D 1.

As before, we can also consider HAL-FI using Bayesian estimation for Hamiltonian learning. In Fig. 10(b), we show results comparing learners with the Bayesian estimator SMC, against learners using alternate estimation. Firstly, we note that in a reversal of fortunes from Fig. 8(b), the passive learner with the Bayesian estimator SMC performs similarly to the passive learner with MLE, and much better than the passive learner with estimation based on linear regression. Secondly, we note that while HAL-FI with SMC outperforms all other learners equipped with Bayesian estimators including the sequential active learner of Qinfer, this advantage has reduced from the case where the query space satisfied the MFC criteria. Lastly, similar scaling behavior is seen for the different learners as earlier.

(c) Experimental dataset. We show a comparison of the performance of the different learners on the *oracle* with access to experimental data in Fig. 11. As expected from our observations on the simulator, we observe SQL like scalings in RMSE with N for the passive learners. We observe around 82.2% query reduction when using the passive learner with the MLE estimator over estimation based on linear regression, similar to that previously observed on the simulator. This query reduction was computed by fixing the RMSE value at 0.2, and comparing the number of queries required by the passive learner to achieve this RMSE value versus the baseline. The trends themselves are remarkably similar to those obtained on the simulator, supporting the fact that the main noise sources affecting the quantum device were identified and the simulator is a good representation of the real quantum hardware.

For HAL-FI, we only show results for fixed query space as the results for the linearly growing query space are very similar, as was also observed on the simulator. We also only show results for HAL-FI with the MLE estimator as HAL-FI produces best results combined with this estimation and we have already shown that HAL-FI is compatible with Bayesian estimation earlier. We see an initial scaling of $\epsilon \sim 1/N^{3/2}$ (or $N \sim \epsilon^{-2/3}$) in RMSE with number of queries, which is higher than SQL and Heisenberg limited scaling, and a scaling of SQL in the asymptotic query regime. The performance of HAL-FI on the experimental data is surprisingly better than that on the simulator. This behavior is consistent across different drive configurations on *ibmq_boeblingen*. The initially accelerated learning where we observe super-Heisenberg limited scaling in HAL-FI and lower values of RMSE achieved for a smaller number of N than on the simulator might be due to: (i) noise sources not included in our model that do not contribute significantly to the noise affecting the quantum device but encourage exploration by HAL-FI, and (ii) correlations between the samples collected in the experimental data (e.g., due to thermal fluctuations).

For the sequential active learner Qinfer, we observe similar behavior in scalings as HAL-FI with an initially accelerated scaling higher than Heisenberg limited scaling and a scaling of SQL in the asymptotic query regime. The RMSE achieved by Qinfer is comparable to HAL-FI for lower values of queries but HAL-FI outperforms Qinfer significantly for higher values of queries (obtaining lower RMSE and with lower uncertainty). This is similar to what was observed on

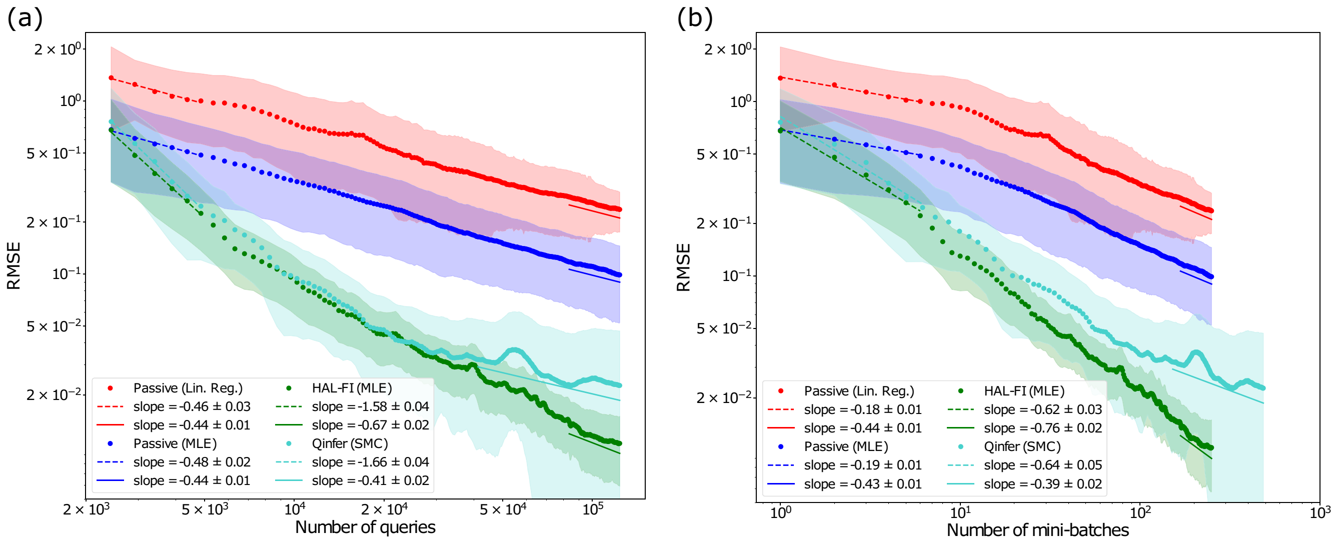

FIG. 11. Scaling of RMSE for different learners on experimental data with (a) number of queries made and (b) number of mini-batches requested. The time range $\mathcal{T}$ of the query space is such that minimum-frequency criteria is not necessarily satisfied for frequency estimation using FFT. We compare the trends in RMSE of HAL-FI using MLE, sequential active learner of Qinfer using Bayes risk as the query criteria and passive learners equipped with different estimators. Estimators of each learner are indicated in brackets. Slopes indicate the scaling of RMSE with number of queries in the finite sample and asymptotic sample regimes. Filled in areas indicate the respective errors on trends for each learner.

the simulator earlier. Moreover, as Qinfer is a sequential active learner, it requests more mini-batches of queries to reach a particular value of RMSE. This is illustrated in Fig. 11(b) where we compare the number of mini-batches of queries made by HAL-FI against Qinfer. Note that the passive learners and HAL-FI request only one mini-batch for every round during learning. A mini-batch in our numerical experiments directly corresponds to a job containing multiple quantum circuits for execution on cloud based IBM quantum devices. Each mini-batch has an associated cost of compilation on the device, and latency from classical electronics interfacing with the quantum hardware. Results of Fig. 11(b) thus indicate that HAL-FI would be preferred over Qinfer on current hardware with a fixed experimental budget.

We have already seen benefits of using HAL-FI over a passive learner and a sequential active learner such as Qinfer through the lower values of RMSE that can be achieved for a given number of queries. This is analyzed in terms of query advantage in Sec. V D 3.

D. Analysis

In this section, we analyze the results of the performance of the different learners from Sec. V C. We firstly comment on the achievability of the Heisenberg limited scaling by HAL-FI with an adaptively growing query space as claimed in Sec. II B. In the process, we consider a different learning scenario motivated by recalibrations of quantum devices. We then describe the query advantage of the active learner under different conditions over the baseline strategy.

1. *Heisenberg limited scaling*

In Sec. V C, we did not observe Heisenberg limited scaling for HAL-FI (even with an adaptively growing query space). This is due to the fact that the query space is not rich enough to achieve Heisenberg limited scaling, i.e., there is no sequence of queries even in the adaptively growing query space to achieve Heisenberg limited scaling. We discuss when Heisenberg limited scaling is achievable for Hamiltonians based on the CR Hamiltonian in Appendix E. We show that the behavior of the learners observed so far is expected through another set of experiments in Appendix D.

It should, however, be possible to achieve Heisenberg limited scaling for a subset of Hamiltonian parameters given the query space (see Sec. IV B 2) when the task is to learn this subset of Hamiltonian parameters and we are given access to information about the other Hamiltonian parameters. This is exactly the setting of a recalibration where prior information about the Hamiltonian parameters is available from previous calibrations and the goal is to learn the subset of parameters, which drift significantly with time while refining estimates of those parameter that do not. Motivated by this, we consider the following learning scenario on *ibm_boeblingen* under drive configuration 3 (see Table IV).

We have access to an estimate of the Hamiltonian parameters $\hat{\boldsymbol{\theta}}$ from a previous calibration during which Hamiltonian learning was run on a uniformly sampled set of queries from $\mathcal{Q}$ (of size $N = 2430$). The goal is to then learn the parameters $\omega_{0,1}$ using the different learners at our disposal. We plot comparisons of different learners (using $N_b = 972$) on this recalibration task in Fig. 12 considering the oracles of the simulator and experimental data. We consider all the learners with estimation based on MLE or linear regression but could also use a Bayesian estimator as we have already noted in Sec. V C. We will call the passive learner with estimation based on linear regression as the baseline for these studies.

We observe the SQL scaling in the baseline, passive learner and HAL-FI in the fixed query space. There is nearly a

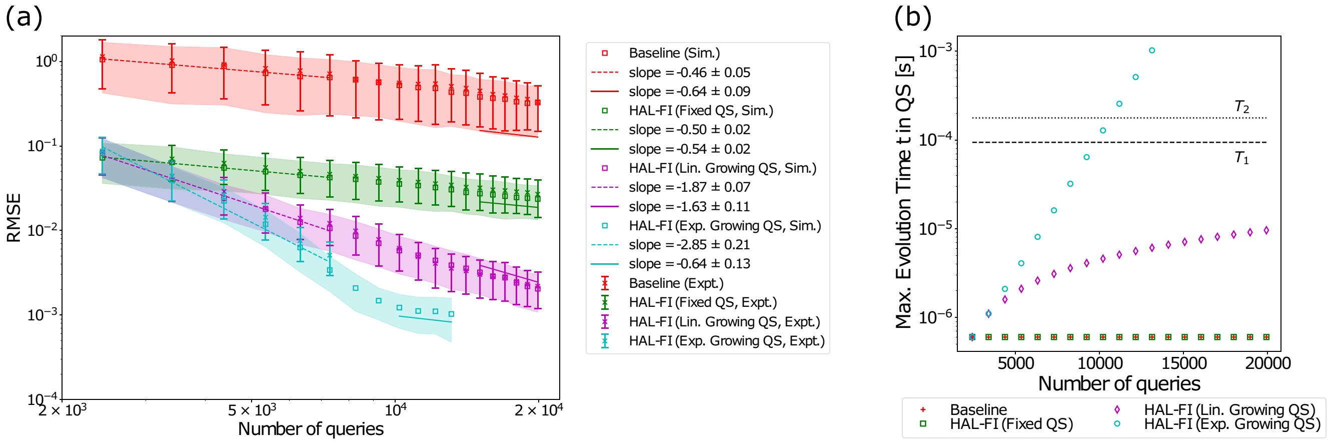

FIG. 12. (a) Trends of RMSE with number of queries for different learners in fixed or growing query spaces (QS) on experimental data (Expt.) compared against simulator (Sim.). Slopes indicate the scaling of RMSE ϵ with number of queries N in the low query/sample and high query/sample regimes. Filled in areas indicate the respective errors on trends for each learner on the simulator. Passive learner, which has similar behavior to HAL-FI in fixed query space is not shown for brevity. (b) Trends of the maximum evolution time associated with query spaces of different learners.

constant gap between the baseline strategy and HAL-FI in the fixed query space, indicating a constant query reduction in achieving a desired learning error. We observe a super-Heisenberg limited scaling of RMSE $\epsilon \sim N^{-3/2}$ in number of queries in HAL-FI with a linearly growing query space. This is also observed for HAL-FI with an exponentially growing query space in the low sample regime. The deterioration in the scaling of HAL-FI with an exponentially growing query space to the SQL is due to the maximum evolution time corresponding to the growing query space eventually exceeding T_1 and T_2. This is shown in Fig. 12(b). In fact, the bend in the trend of RMSE versus N for HAL-FI with an exponentially growing query space occurs immediately after the maximum evolution time in $\mathcal{T}$ exceeds T_1. As $\mathcal{T}$ for HAL-FI with a linearly growing query space is grown much more slowly, effects of decoherence are not yet felt and super-Heisenberg limited scaling convergence rate in learning error is achieved.

Qualitatively, it is clear that much lower values of learning error can be achieved with a given budget of queries using HAL-FI with an adaptively growing query space over the baseline for recalibration. This is quantified in terms of query advantage in Sec. V D 3.

2. Computational cost

So far, we have assessed the performance of different learners in achieving a value of RMSE with respect to the number of queries consumed. However, another relevant resource is the classical computational time required for learning the Hamiltonian parameters during estimation or that required in computing query criteria for active learners.

The main contribution to this classical computational time is the evaluation of the likelihood function for different queries, required during estimation and for evaluation of query criteria for active learning. If an analytical expression for the likelihood function is not available, then one needs to carry out expensive quantum simulations. The computational cost of using the SMC estimator in a round during learning scales as $O(|\mathcal{Q}|n_p)$ with the number of particles. On the other hand, the cost (or number of likelihood evaluations) of using MLE in a round scales as $O(|\mathcal{Q}|n_{\text{iters}})$ with the number of iterations n_{iters} within the estimation procedure (e.g., number of gradient steps, etc.). In HAL-FI, we also account for the number of likelihood evaluations in setting up the Fisher information matrices of different queries for solving the SDP of Eq. (19). In Qinfer, we account for the number of likelihood calls in hypothetical updates of the risk particles for evaluating Bayes risk (see Sec. V B). In Fig. 13, we compare the RMSE achieved for increasing computational cost for different learners equipped with the MLE estimator or the Bayesian SMC estimator. Overall, we find that learners equipped with MLE are less computationally expensive compared to those equipped with the Bayesian estimator.

In our experiments, we observe that the number of iterations (e.g., of the quasi-Newton solve) in the estimation procedure of MLE (Sec. IV D 2) is lower than the number of particles n_p required in SMC. This translates into about two orders of magnitudes of savings in computational cost when comparing HAL-FI against the sequential active learner of Qinfer on both the simulator and experimental data.

3. Query advantage

We have so far compared the trends and obtained the scalings of RMSE ϵ with number of queries N for different learners for Hamiltonian learning (Sec. V C) and with prior information (Sec. V D 1). To quantify the benefits of using HAL-FI over other learners, we now evaluate its query advantage (QA), defined in Sec. II C 4 as a performance metric that summarizes the reduction in resources required to achieve a desired learning error. We restrict our attention to HAL-FI with the MLE estimator to simplify the discussion.

We plot the trend of QA as a function of RMSE for active learners HAL-FI and Qinfer over baselines of passive learners equipped with different estimators as achieved on the simulator in Fig. 14 and as achieved on experimental

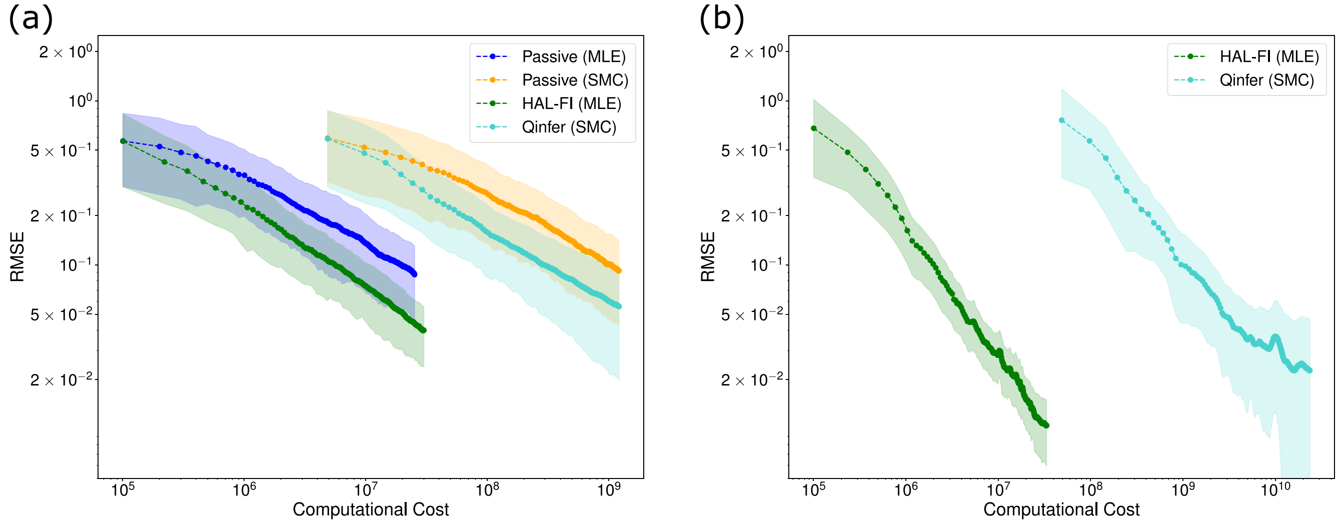

FIG. 13. Trends of RMSE with computational cost for different learners on (a) simulator where MFC is not necessarily satisfied, and (b) experimental data. The computational cost is measured in terms of the number of likelihood evaluations during learning.

data in Fig. 15(a). Values of QA for certain values of RMSE are tabulated in Tables V and VI for different combinations of learners and baselines. We observe a QA of around 80% in HAL-FI on the simulator and experimental data over the baseline of the passive learner with estimation based on linear regression, for high values of RMSE. The initial accelerated learning observed for HAL-FI on the simulator in Fig. 10 and experimental data in Fig. 11 translates to an accelerated QA for high values of RMSE. The trend in QA for HAL-FI over the passive learner with estimation based on linear regression on the simulator flattens to an asymptotic value of at least 95.1% when the query space is fixed during learning and at least 96.3% when the query space is grown linearly. The corresponding value for HAL-FI on the experimental data in the fixed query space is 99.8% for low values of RMSE. Similarly, the QA of HAL-FI over a passive learner with the MLE estimator is around 82.1% on the simulator and around 99.0% on experimental data for low values of RMSE. In comparison, the sequential active learner of Qinfer is only able to achieve a QA of around 62.7% on the simulator but performs better on the experimental data achieving a QA of around 96.9% over the baseline of passive learner with the MLE estimator.

For QA of other combinations of learners and baselines, we turn our attention to Tables V and VI. We note that HAL-FI can achieve up to a QA of 53.2% and 70.9% in expectation over the baseline of Qinfer on the simulator and experimental data respectively for the lowest value of RMSE achieved by HAL-FI in our numerical experiments in Fig. 11. There is, however, a large uncertainty on the estimate of QA on the experimental due to the fluctuations of the trend of Qinfer on the experimental data (Fig. 11). In practice, we expect that

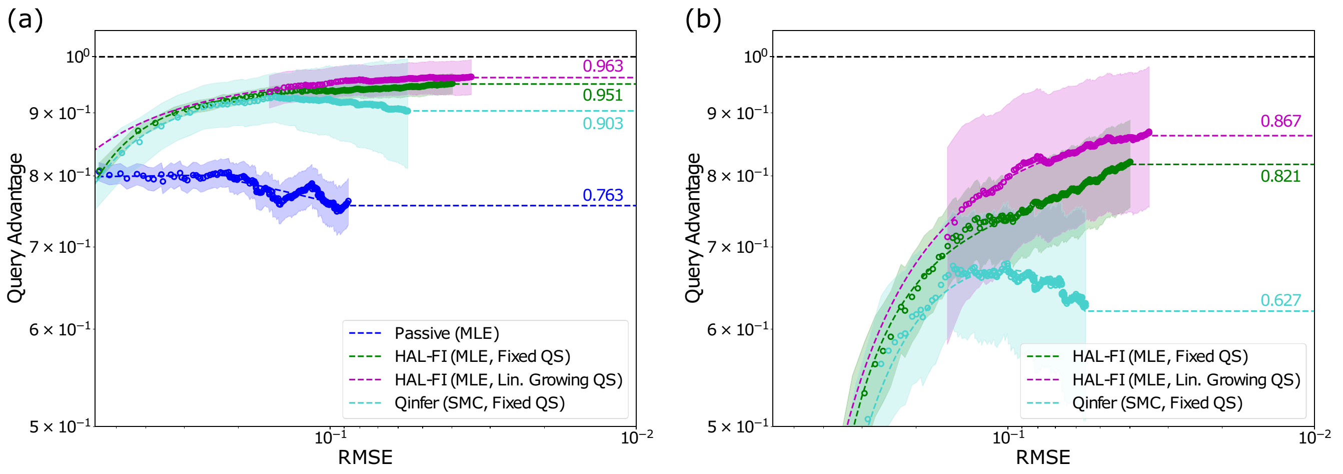

FIG. 14. Query advantage (QA) of different learners on the simulator when MFC (minimum-frequency criteria) is not satisfied over the following baselines: (a) passive learner with estimation based on linear regression and (b) passive learner with the MLE estimator. In (a) and (b), the data points correspond to values of QA computed from the data points of Fig. 10 and the lines are the fits to these values of QA. The annotated text for the different trends indicate the QA obtained for the corresponding learner over the baseline for the lowest observed value of RMSE in our numerical experiments. Filled in areas indicate the 95% confidence interval on the data points shown.

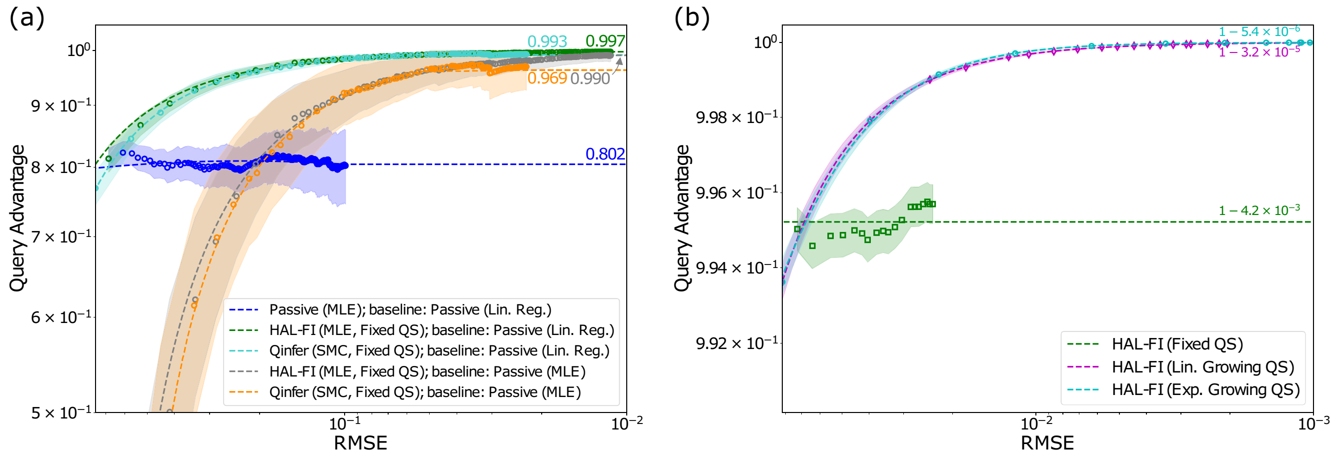

FIG. 15. Query advantage (QA) of different learners over specified baselines considering the problem of (a) Hamiltonian learning without prior information and (b) Hamiltonian learning with prior information on subset of Hamiltonian parameters from previous calibrations. In (a), we plot the QA of passive learners, HAL-FI, and Qinfer on experimental data over two different baselines. In (b), we plot the QA of HAL-FI over the baseline of the passive learner on the simulator for fixed and adaptively growing query spaces (QS). In (b), the QAs obtained on the simulator closely match those obtained on the experimental data and hence the latter is not shown. In (a) and (b), the data points correspond to values of QA computed from the data points of Figs. 11 and 12 and the lines are the fits to these values of QA. The annotated text for the different trends indicate the QA obtained for the corresponding learner over the baseline for the lowest observed value of RMSE in our numerical experiments. Filled in areas indicate the 95% confidence interval on the data points shown.

TABLE V. Query advantage of various learners considering different baselines on a simulator when MFC is not satisfied. Estimators of each learner are indicated in brackets. For each combination of learner and baseline, we report query advantage at lowest value of RMSE achieved by the learner (compared to extrapolated fit of baseline) and the query advantage at lowest value of RMSE achieved by the baseline in brackets below. The errors represent one standard deviation on the value of expected query advantage.

Learner Baseline	Passive (MLE)	Qinfer (SMC)	HAL-FI (MLE, Fixed QS)	HAL-FI (MLE, Lin. Growing QS)
Passive (Lin. Reg.)	0.763 ± 0.034 (0.788 ± 0.024)	0.903 ± 0.092 (0.926 ± 0.043)	0.951 ± 0.018 (0.929 ± 0.011)	0.963 ± 0.031 (0.935 ± 0.029)
Passive (MLE)	×	0.627 ± 0.122 (0.668 ± 0.090)	0.821 ± 0.067 (0.750 ± 0.062)	0.867 ± 0.113 (0.811 ± 0.114)
Qinfer (SMC)	×	×	0.532 ± 0.174 (0.416 ± 0.182)	0.658 ± 0.289 (0.572 ± 0.302)
HAL-FI (MLE, Fixed QS)	×	×	×	0.184 ± 0.091 (0.215 ± 0.081)

TABLE VI. Query advantage (QA) of various learners considering different baselines on experimental data. Estimators of each learner are indicated in brackets. For each combination of learner and baseline, we report query advantage at lowest value of RMSE achieved by the learner (compared to extrapolated fit of baseline) and the query advantage at lowest value of RMSE achieved by the baseline in brackets below. The errors represent one standard deviation on the value of expected query advantage and are underlined in cases where the range of query advantage should be read as [QA-error,1) instead of [QA-error, QA+error].

Learner Baseline	Passive (MLE)	Qinfer (SMC)	HAL-FI (MLE, Fixed QS)
Passive (Lin. Reg.)	0.802 ± 0.057 (0.809 ± 0.034)	0.993 ± 0.006 (0.953 ± 0.012)	0.997 ± 0.001 (0.957 ± 0.009)
Passive (MLE)	×	$0.969 \pm \underline{0.046}$ (0.917 ± 0.056)	$0.990 \pm \underline{0.011}$ (0.916 ± 0.046)
Qinfer (SMC)	×	×	$0.709 \pm \underline{0.373}$ (0.510 ± 0.169)

HAL-FI would achieve a QA upwards of 33% (computed by considering one standard deviation below the expected value of QA).

Similarly for the learning scenario of recalibration, we plot the QA of HAL-FI over the baseline of passive learner (with estimation based on linear regression or MLE) for Hamiltonian learning when prior information is available in Fig. 15(b). The trends shown here correspond to that of the simulator as similar behavior is observed for experimental data. The QA of HAL-FI in a fixed query space over the baseline is constant with RMSE and around 99.6%, i.e., HAL-FI requires only 0.4% of the queries required by the baseline. For HAL-FI with a linearly growing query space, the query advantage for low values of reported RMSE is around $(100 - 3.2 \times 10^{-3})\%$. Further, we note that the scaling of QA of HAL-FI in a linearly growing query space with RMSE ϵ scales as QA $\sim (1 - O(\epsilon^{-1/6}))$ until decoherence starts effecting the scaling. For HAL-FI with an exponentially growing query space, the QA flattens to around $(100 - 5.4 \times 10^{-4})\%$ for low values of RMSE. Under this learning setting at low values of RMSE around 10^{-3}, HAL-FI has a query space with evolution times far exceeding T_1 or T_2 and thus this reported QA is expected for even lower values of RMSE.

From the analysis of query advantage above, we observe that two orders of magnitude reduction in queries can be obtained over a baseline of passive learner with estimation based on linear regression by adopting HAL-FI. Moreover, during recalibrations, we observe three orders of magnitude reduction when HAL-FI is used with a linearly growing query space and five orders of magnitude reduction in queries when HAL-FI is used with an exponentially growing query space.

Adopting HAL-FI not only allows us to achieve query reduction but it also allows us to save wall clock time taken to calibrate a quantum device. We again consider the query space of Sec. IV B 2 where a query on average takes 2400 ns to run (accounting for time duration of implementing measurement pulses). The repetition rate of current IBM quantum devices for executing circuits is 10 kHz. We can then reduce the duration of Hamiltonian learning of all CR Hamiltonians of directly connected qubit pairs on a 20-qubit IBM quantum device of *ibmq_boeblingen* to reach a RMSE of 5×10^{-2} from around 10 minutes to 5 seconds by using HAL-FI instead of a passive learner with estimation based on regression [71]. These timings only take into account the time on the quantum hardware and not additional latencies in classical electronics interfacing with the hardware.

VI. CONCLUSIONS

In this paper, we proposed the active learning algorithms of HAL-FI for Hamiltonian learning and HAL-FIR for predictions of queries to a Hamiltonian, sampled from a testing distribution. They both perform in batch-mode making them ideal for use on near-term quantum hardware. The performance of HAL-FI/HAL-FIR was compared against different learners for learning a CR Hamiltonian on the 20-qubit IBM quantum device *ibmq_boeblingen* on a simulator and experimental data. We showed that HAL-FI can achieve a query advantage of around 99.7% over a passive learner with estimation based on linear regression, 99.0% over a passive learner with MLE estimator, and upwards of 33% over the sequential active learner of Qinfer (equipped with the query criteria of Bayes risk) for low values of learning error on experimental data. Moreover, HAL-FI is able to achieve the same learning error at lower computational cost than passive learners and Qinfer. During recalibration when learning the Hamiltonian with access to information from previous calibrations, we observed that HAL-FI can achieve query advantages of 99.5% over passive learners. Further, we showed that we achieve Heisenberg limited rate of convergence where possible when an active learner is used in conjunction with an adaptive query space during learning before the evolution time of queries exceed qubit T_1 or T_2 and decoherence starts deteriorating information content available in queries.

Overall, using the active learner HAL-FI operating in batch mode can yield a reduction in resources of up to two orders of magnitude during calibration and five orders of magnitude during recalibration for low values of learning error. This improvement in query complexity has multiple practical consequences besides accelerating Hamiltonian learning during calibration and recalibrations of quantum computers.

Another important calibration step is determining controls to implement desired single and multi-qubit quantum gates. This often relies on building a Hamiltonian model of the true environment, i.e., the quantum computer. This can be accomplished by HAL to ensure minimal queries are used. Gates, once implemented, are characterized through quantum process tomography, which is not query efficient but could be accelerated with an active learner. For the specific application of learning the CR Hamiltonian, more noise sources can be included as they become relevant, e.g., leakage errors, which become pronounced under strong driving. It would also be interesting to see if one can achieve asymptotic Heisenberg limited scaling even in the presence of noise such as decoherence by using appropriate quantum error correction protocols. It should be noted that Hamiltonian learning and for that matter quantum process tomography both suffer from an exponential scaling in the size of the quantum device n and using an active learner only ensures a better scaling in ϵ or query advantage. One could get a better scaling in n if additional information such as the structure of the Hamiltonian was known to be a k-local Hamiltonian, which can be learned with a query complexity that scales as $O(\text{poly}(n))$ [85].

There are other calibration steps where Hamiltonian learning may not be required but the concept of active learner can be introduced. This would particularly be advantageous where a variety of experiments can be carried out but it is not clear which of them are more informative for the learning task. For example, a query efficient method is desired for learning cross-talk on a superconducting quantum device [86,87].

Additionally, there is room for improvement and extension in the algorithm itself. Currently, expert knowledge is required to specify a complete query space to HAL to ensure all the Hamiltonian parameters can be learned and possibly with Heisenberg limited scaling. It is desirable to remove this expert and replace them with a method to synthesize queries. The current active learning strategy is also based on the query criterion of Fisher information but this could be modified to incorporate cost of different queries and incentivize exploration. Moreover, a more general query criteria could possibly

TABLE VII. Relevant parameters of IBM quantum devices. The qubit used as the control or target qubit is indicated by its number in the device connectivity map shown in Fig 16. Error per gate refers to the average infidelity of single-qubit gates implemented on that qubit.

	Qubit		Qubit freq. (GHz)		T_1 (μs)		T_2 (μs)		Error per gate ($\times 10^{-4}$)	
Device	Control	Target	Control	Target	Control	Target	Control	Target	Control	Target
A	1	0	5.0593	4.8441	65.4 ± 10.8	32.9 ± 9.5	71.1 ± 5.9	57.7 ± 9.2	5.65 ± 0.13	8.85 ± 0.46
B	3	0	5.1482	4.9273	63.2 ± 11.7	78.1 ± 24.9	73.9 ± 9.9	124.3 ± 21.5	5.92 ± 0.30	6.29 ± 0.23
C	0	1	5.3613	5.2910	34.2 ± 2.5	45.0 ± 19.1	39.2 ± 3.3	63.1 ± 7.6	18.3 ± 1.0	20.6 ± 1.3
	0	2	5.3613	5.2543	34.2 ± 2.5	45.0 ± 19.1	35.8 ± 0.7	52.4 ± 4.2	18.3 ± 1.0	9.23 ± 0.26
	1	2	5.2910	5.2543	39.2 ± 3.3	63.1 ± 37.6	35.8 ± 0.7	52.4 ± 4.2	20.6 ± 1.3	9.23 ± 0.26
D	0	1	5.0466	4.8468	94.0 ± 6.0	75.7 ± 17.0	177.2 ± 44.8	128.1 ± 29.7	2.39 ± 0.12	3.12 ± 0.11

be learned through reinforcement learning as illustrated in recent work on classical applications [88].

Code for the passive learner and the active learner HAL using estimators based on regression and MLE is available at [89]. Code for the learners including Qinfer for the cross-resonance Hamiltonian, all equipped with a Bayesian estimator is available at [90]. Datasets generated from the simulator are included in the above repositories. Please contact the corresponding author for access to the experimental datasets collected from the IBM quantum devices and used in the paper.

ACKNOWLEDGMENTS

A.D. was supported in part by the MIT-IBM Watson AI Lab. A.D., J.S., and I.L.C. were supported in part by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-Design Center for Quantum Advantage under Contract No. DE-SC0012704. A.D. acknowledges the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing computational resources. The authors thank the anonymous referees for their helpful comments that greatly improved the first version of this paper.

APPENDIX A: DETAILS OF CROSS-RESONANCE HAMILTONIAN

In this Appendix, we firstly give details of the different IBM quantum devices employed for assessing the performance of the HAL algorithms (HAL-FI and HAL-FIR) proposed in Sec. III for learning cross-resonance (CR) Hamiltonians [Eq. (31)]. In Appendix A 2, we discuss how the queries from Sec. IV B 2 are implemented in practice on the devices. In Sec. IV C 2, we stated a model for the noise source of imperfect pulse-shaping. Here in Appendix A 3, we describe how this model was obtained. This is then followed by relevant analytical expressions of likelihood and Fisher information in Appendix A 4 considering the query space in Sec. IV B 2.

1. Description of IBM quantum devices

We consider CR Hamiltonians on the four different IBM quantum devices described in Sec. IV B 1. The connectivity maps of these devices are shown in Fig. 16. We consider CR gates on particular qubit pairs on each device, which are summarized in Table VII. In Table VII, we describe the properties of each qubit involved in the CR gate including their T_1 or T_2 times and the average infidelity of single-qubit gates.

2. Experimental implementation of query space

Queries to the CR Hamiltonians between different qubit pairs on the IBM quantum devices are made through appropriate pulse sequences. These pulse sequences are constructed and executed on the hardware using `Qiskit-Pulse` [75], which is a pulse programming module within `Qiskit` [76] and serves as a front-end implementation of the `OpenPulse` interface [91]. Each `Qiskit-Pulse` [75] program consists of pulses, channels and instructions. Here, we describe a `Qiskit-Pulse` program and describe how a query to a CR gate on an IBM quantum device is specified.

A pulse is a time-series of complex-valued amplitudes with maximum unit norm and which we denote as a_k where $k \in [n-1]$ corresponds to the time stamp. The difference between

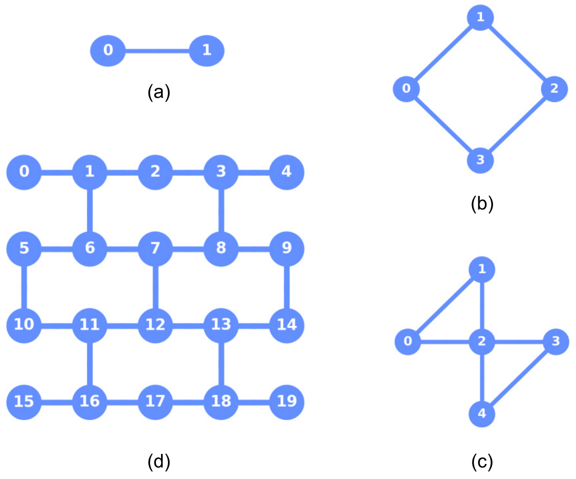

FIG. 16. Connectivity maps for (a) IBM quantum device A, (b) IBM quantum device B, (c) IBM quantum device C, and (d) IBM quantum device D *ibmq_boeblingen*. Each node represents a physical qubit on the chip and the presence of an edge between two nodes in the connectivity map indicates that a CR gate can be applied between these two nodes.

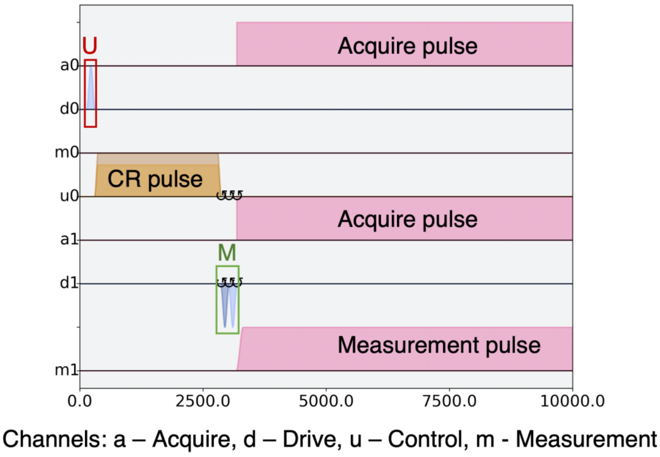

FIG. 17. An example of a CR pulse schedule on the IBM quantum device *ibmq_boeblingen* considering to the query of $x = (M, U, t)$ where $M = \sigma_I \otimes \exp(-i\frac{\pi}{4}\sigma_X)$, $U = \sigma_X \otimes \sigma_I$, and time duration $t = 6 \times 10^{-7}$ s. The x axis corresponds to time normalized by $dt = 2.22 \times 10^{-10}$ [Eq. (A1)]. The different channels corresponding to each qubit (y axis) are written as the type of channel (see plot legend) followed by qubit number. Qubit 0 is set to be the control qubit and 1 to be the target qubit. The envelope of the different pulses are shown in each channel. The rotations on the drive or control channels indicate virtual Z gates. An equivalent representation of the quantum circuit is shown in Fig. 1.

these time stamps is considered to be dt, which is typically the sample rate of the waveform generator. The output signal thus has an amplitude of

$$A_k = \mathrm{Re}[e^{i2\pi fkdt+\gamma} a_k] \tag{A1}$$

at time kdt where f and γ are a modulation frequency and phase. A pulse is specified in `Qiskit-Pulse` by specifying the individual amplitudes a_k and the phase ϕ. Alternatively, one can use parametric pulse shapes that are offered by the library such as `Gaussian`, `GaussianSquare`, etc. These pulses are then implemented on the hardware via channels, which label signal lines used for transmitting and receiving signals between the control electronics, and the hardware. In particular, these are implemented on the `PulseChannel` and are used to control the system Hamiltonian to implement different gates.

To implement a query to a quantum device, we need an equivalent description of the quantum circuit shown in Fig. 1 in the form of a pulse schedule. We discuss this here in parallel with a description of the query space $\mathcal{Q}$ considered for learning CR Hamiltonians. To prepare the initial state, we consider the set of preparation operators $\mathcal{U} = \{\sigma_I \otimes \sigma_I, \sigma_X \otimes \sigma_I\}$ applied to the pure state $|00\rangle$. The single-qubit gates are implemented as a sequence of Gaussian pulses of the appropriate amplitude and duration [75]. Assuming the first (left) qubit is the control and the second (right) qubit is the target, the effect of the preparation operators is to place the control in $|0\rangle$ and $|1\rangle$ respectively. We evolve the initial state $|\psi(0)\rangle$ for time $t \in \mathcal{T}$, which we will specify when discussing the results of our application of Hamiltonian learning in Sec. V. This is done by switching the CR interaction on for time duration t, which is done in practice by implementing a `GaussianSquare` pulse of duration t. A `GaussianSquare` pulse is a square pulse with truncated Gaussian-shaped rising and falling edges. We discuss later in Sec. IV C how using this pulse may introduce nonidealities in the system evolution. Finally after obtaining the final state $|\psi(t)\rangle$, we apply the measurement operators in $\mathcal{M} = \{\sigma_I \otimes \exp(i\frac{\pi}{4}\sigma_Y), \sigma_I \otimes \exp(-i\frac{\pi}{4}\sigma_X), \sigma_I \otimes \sigma_I\}$ and measure only the second qubit, which we have chosen as the target qubit. The query space is then $\mathcal{Q} = \mathcal{M} \times \mathcal{U} \times \mathcal{T}$. An example of a pulse schedule is shown in Fig. 17 highlighting the different parts of the query. Moreover, in our experimental setup, we obtain measurements of the single-shot signal (integrated cavity amplitude) $\mathbf{c}$, which is a function of the measurement outcomes $\mathbf{y}$, which we have described earlier.

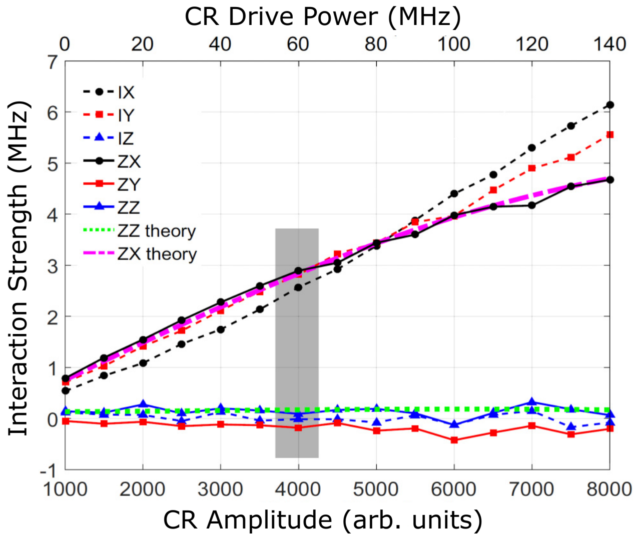

FIG. 18. Hamiltonian parameters **J** as a function of the amplitude of the control pulse or drive as reported in Sheldon *et al.* [71].

3. Modeling pulse shapes

We now describe how the imperfect pulse-shaping model stated in Sec. IV C 2 was obtained. We consider cross-resonance control pulses (also called `GaussianSquare`) whose time-varying amplitudes are rectangular-shaped envelopes with tapered rising and falling edges, where the tapering is designed to minimize the signal energy that falls above and below the frequency of the sinusoid that is being modulated by the pulse envelope. The resulting unitary operators thus have the form $\tilde{U}(t) = \mathbb{T}\exp\big(-i\int_0^t \tilde{H}(t')dt'\big)$, where $\mathbb{T}$ is the time ordering operator, $\tilde{H}$ is the Hamiltonian at any particular time given by $\tilde{H}(t') = H(t', v(t'))$, H is the cross-resonance Hamiltonian (with the dependence on parameters $\boldsymbol{\theta}$ not shown), and $v(t')$ is the time-varying pulse envelope.

Let Δt_r and Δt_f be, respectively, the durations of the rising and falling edges of the shaped pulse envelope. The central portion of $v(t')$ is then a rectangular function such that, for $t' \in [\Delta t_r, t - \Delta t_f]$, $v(t') = V_{\max}\mathbb{1}_{t' \in [\Delta t_r, t-\Delta t_f]}$ where t is the total duration of the pulse, and $V_{\max}$ is the amplitude of this central rectangular portion. We thus have

$$\begin{aligned}\tilde{U}(t) &= \exp\left(-i\mathbb{T}\int_0^t \tilde{H}(t')dt'\right) &\text{(A2)}\\ &= \exp\left(-i\mathbb{T}\int_{t-\Delta t_f}^{t} H(t', v(t'))dt'\right)\exp\left(-i\mathbb{T}\int_{\Delta t_r}^{t-\Delta t_f} H(t', V_{\max})dt'\right)\exp\left(-i\mathbb{T}\int_0^{\Delta t_r} H(t', v(t'))dt'\right) &\text{(A3)}\\ &= \exp\left(-i\mathbb{T}\int_{t-\Delta t_f}^{t} H(t', v(t'))dt'\right)\exp\left(-iH_{\max}\int_{\Delta t_r}^{t-\Delta t_f} dt'\right)\exp\left(-i\mathbb{T}\int_0^{\Delta t_r} H(t', v(t'))dt'\right) &\text{(A4)}\\ &= \exp\left(-i\mathbb{T}\int_{t-\Delta t_f}^{t} H(t', v(t'))dt'\right)\exp\left(-iH_{\max}t_{\text{expt}}\right)\exp\left(-i\mathbb{T}\int_0^{\Delta t_r} H(t', v(t'))dt'\right) &\text{(A5)}\end{aligned}$$

where $t_{\text{expt}} = t - \Delta t_f - \Delta t_r$ and $H_{\max} = H(t', V_{\max})$, which is constant assuming that any signal distortions that are introduced by the control electronics and/or along the signal path to the quantum device are negligible.

The above equation decomposes $\tilde{U}(t)$ into the time evolution of the Hamiltonian that corresponds to the central rectangular-pulse portion of the control pulse with pre- and post-rotations that are determined by the tapered rising and falling edges of the pulse. In general, there is a nonlinear relationship between the shapes of these edges and the resulting pre- and post-rotations. However, based on the results reported in [71] and shown in Fig. 18, the cross-resonance Hamiltonian parameters tend to vary fairly linearly with respect to the overall pulse amplitude. The pre- and post-rotations can thus be approximated by assuming a first-order model for the time-varying Hamiltonian parameters given by

$$\mathbf{J}(t') \approx \frac{v(t')}{V_{\max}}\mathbf{J}_{\max} \tag{A6}$$

where $\mathbf{J}_{\max}$ is the vector of parameters for the Hamiltonian $H_{\max}$ of the central recetangular portion of the pulse envelope. The time-varying Hamiltonian is then approximated by

$$H(t', v(t')) \approx \frac{v(t')}{V_{\max}}H_{\max}. \tag{A7}$$

The overall unitary operator $\tilde{U}(t)$ is then approximated by

$$\begin{aligned}\tilde{U}(t) &\approx \exp\left(-iH_{\max}\frac{1}{V_{\max}}\int_{t-\Delta t_f}^{t} v(t')dt'\right)\\ &\quad\times\exp\left(-iH_{\max}t_{\text{expt}}\right)\exp\left(-iH_{\max}\frac{1}{V_{\max}}\int_0^{\Delta t_r} v(t')dt'\right) &\text{(A8)}\\ &= \exp\left(-iH_{\max}(t_{\text{expt}} + \Delta t_{\text{eff}})\right) &\text{(A9)}\end{aligned}$$

where

$$\Delta t_{\text{eff}} = \frac{1}{V_{\max}}\left(\int_0^{\Delta t_r} v(t')dt' + \int_{t-\Delta t_f}^{t} v(t')dt'\right). \tag{A10}$$

4. Likelihood function and Fisher information matrix for the CR Hamiltonian

In this section, we give the expressions for the likelihood function of $p_{\mathrm{y|x}}(y|x;\boldsymbol{\theta})$ and the Fisher information (FI) matrix. We consider the experimental setup as described in Sec. IV and query space $\mathcal{Q}$ as described in Sec. IV B 2.

Recall from Sec. II C 2, the FI matrix of a query x is given by

$$\mathcal{I}_x(\boldsymbol{\theta})[i,j] = \mathbb{E}\left[\frac{\partial \log p_{\mathrm{y|x}}(y|x;\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial \log p_{\mathrm{y|x}}(y|x;\boldsymbol{\theta})}{\partial\theta_j}\right] \tag{A11}$$

where $\log p(y|x;\boldsymbol{\theta})$ is the log-likelihood of the measurement outcome y given the query x. In most cases in practice, the Fisher information matrix must be computed empirically in a Monte Carlo fashion. However, here we have a model of the CR Hamiltonian and models of the different noise sources affecting the quantum system available to us. We can thus evaluate the FI matrix analytically for different queries and in the presence or absence of noise.

a. In Absence of Noise

(a) Likelihood. In the noiseless case, the likelihood function of different measurement outcomes $y \in \{0, 1\}$ given query $x = (M, U, t) \in \mathcal{Q}$ is

$$p_{\mathrm{y|x}}(y|x;\boldsymbol{\theta}) = \sum_{z\in\{0,1\}} |\langle yz|Me^{-iH(\boldsymbol{\theta})t}U|00\rangle|^2. \tag{A12}$$

Evaluating this for the different queries in the query space as described in Sec. IV B 2, we obtain

$$p_{\mathrm{y|x}}(0|x;\boldsymbol{\theta}) = \begin{cases}\frac{1}{2}((\cos(\omega_j t) + \sin(\phi_j)\cos(\delta_j)\sin(\omega_j t))^2 + (\sin(\delta_j)\sin(\omega_j t) + \cos(\phi_j)\cos(\delta_j)\sin(\omega_j t))^2), & x = (M_{\langle X\rangle}, U_j, t)\\ \frac{1}{2}((\cos(\omega_j t) - \cos(\phi_j)\cos(\delta_j)\sin(\omega_j t))^2 + (\sin(\delta_j)\sin(\omega_j t) + \sin(\phi_j)\cos(\delta_j)\sin(\omega_j t))^2), & x = (M_{\langle Y\rangle}, U_j, t)\\ 1 - (\cos(\delta_j)\sin(\omega_j t))^2, & x = (M_{\langle Z\rangle}, U_j, t)\end{cases} \tag{A13}$$

where we have used the index $j \in \{0, 1\}$ to refer to the different preparation operators $U_0 = \sigma_I \sigma_I$ and $U_1 = \sigma_X \sigma_I$. The measurement operators are: $M_{\langle X \rangle} = \sigma_I \otimes \exp(i\frac{\pi}{4}\sigma_Y)$, $M_{\langle Y \rangle} = \sigma_I \otimes \exp(-i\frac{\pi}{4}\sigma_X)$, and $M_{\langle Z \rangle} = \sigma_I \otimes \sigma_I$.

(c) Fisher information matrix. Noting that the measurement outcome $y \in \{0, 1\}$, we have

$$\mathcal{I}_x(\boldsymbol{\theta})[i, j] = \sum_{y \in \{0,1\}} \frac{1}{p_{\mathrm{y|x}}(y|x;\boldsymbol{\theta})} \frac{\partial p_{\mathrm{y|x}}(y|x;\boldsymbol{\theta})}{\partial \theta_i} \frac{\partial p_{\mathrm{y|x}}(y|x;\boldsymbol{\theta})}{\partial \theta_j} \tag{A14}$$

$$= \frac{1}{p_{\mathrm{y|x}}(0|x;\boldsymbol{\theta})(1 - p_{\mathrm{y|x}}(0|x;\boldsymbol{\theta}))} \frac{\partial p_{\mathrm{y|x}}(0|x;\boldsymbol{\theta})}{\partial \theta_i} \frac{\partial p_{\mathrm{y|x}}(0|x;\boldsymbol{\theta})}{\partial \theta_j} \tag{A15}$$

where in the second step, we have used the fact that $p_{\mathrm{y|x}}(1|x;\boldsymbol{\theta}) = 1 - p_{\mathrm{y|x}}(0|x;\boldsymbol{\theta})$ and $\frac{\partial p_{\mathrm{y|x}}(1|x;\boldsymbol{\theta})}{\partial \theta_i} = -\frac{\partial p_{\mathrm{y|x}}(0|x;\boldsymbol{\theta})}{\partial \theta_i}$.

The FI matrix elements can also be expressed using the Rabi oscillation of a query $p_{\mathrm{rabi}}(x;\boldsymbol{\theta})$ as follows:

$$\mathcal{I}_x(\boldsymbol{\theta})[i, j] = \frac{1}{1 - p_{\mathrm{rabi}}^2(x;\boldsymbol{\theta})} \frac{\partial p_{\mathrm{rabi}}(x;\boldsymbol{\theta})}{\partial \theta_i} \frac{\partial p_{\mathrm{rabi}}(x;\boldsymbol{\theta})}{\partial \theta_j}. \tag{A16}$$

The FI matrices $\mathcal{I}_x(\boldsymbol{\theta})$ for the different queries $x = (M, U, t)$ (Sec. IV B 2) depend on the parametrization of choice. We denote the FI matrix considering the parametrization of $\boldsymbol{\Lambda}$ as $\mathcal{I}_x(\boldsymbol{\Lambda})$. Note that the Fisher information matrices $\mathcal{I}_x(\mathbf{J})$ for the parametrization of $\mathbf{J}$ is related to the former through the jacobian of $\boldsymbol{\Lambda}$ with respect to $\mathbf{J}$,

$$D_{\boldsymbol{\Lambda},\mathbf{J}} = \begin{bmatrix} \frac{\mathrm{Re}(\beta_0)}{\omega_0} & -\frac{\mathrm{Re}(\beta_0)a_0}{|\beta_0|a_0} & -\frac{\mathrm{Im}(\beta_0)}{|\beta_0|^2} & \frac{\mathrm{Re}(\beta_1)}{\omega_1} & -\frac{\mathrm{Re}(\beta_1)a_1}{|\beta_1|a_1} & -\frac{\mathrm{Im}(\beta_1)}{|\beta_1|^2} \\ \frac{\mathrm{Im}(\beta_0)}{\omega_0} & -\frac{\mathrm{Im}(\beta_0)a_0}{|\beta_0|a_0} & \frac{\mathrm{Re}(\beta_0)}{|\beta_0|^2} & \frac{\mathrm{Im}(\beta_1)}{\omega_1} & -\frac{\mathrm{Im}(\beta_1)a_1}{|\beta_1|a_1} & \frac{\mathrm{Re}(\beta_1)}{|\beta_1|^2} \\ \frac{a_0}{\omega_0} & \frac{|\beta_0|}{\omega_0^2} & 0 & \frac{a_1}{\omega_1} & \frac{|\beta_1|}{\omega_1^2} & \\ \frac{\mathrm{Re}(\beta_0)}{\omega_0} & -\frac{\mathrm{Re}(\beta_0)a_0}{|\beta_0|a_0} & -\frac{\mathrm{Im}(\beta_0)}{|\beta_0|^2} & -\frac{\mathrm{Re}(\beta_1)}{\omega_1} & \frac{\mathrm{Re}(\beta_1)a_1}{|\beta_1|a_1} & \frac{\mathrm{Im}(\beta_1)}{|\beta_1|^2} \\ \frac{\mathrm{Im}(\beta_0)}{\omega_0} & -\frac{\mathrm{Im}(\beta_0)a_0}{|\beta_0|a_0} & \frac{\mathrm{Re}(\beta_0)}{|\beta_0|^2} & -\frac{\mathrm{Im}(\beta_1)}{\omega_1} & \frac{\mathrm{Im}(\beta_1)a_1}{|\beta_1|a_1} & -\frac{\mathrm{Re}(\beta_1)}{|\beta_1|^2} \\ \frac{a_0}{\omega_0} & \frac{|\beta_0|}{\omega_0^2} & 0 & -\frac{a_1}{\omega_1} & -\frac{|\beta_1|}{\omega_1^2} & \end{bmatrix} \tag{A17}$$

where the (i, j)th element is given by $\partial \Lambda_j / \partial J_i$ with $a_{0,1}$ and $\beta_{0,1}$ as defined as in Eq. (31).

Note that $\mathcal{I}_x(\boldsymbol{\Lambda})$ is of rank-1 for each query and takes a block form of $[\begin{smallmatrix}\mathcal{I}_0 & 0 \\ 0 & 0\end{smallmatrix}]$ for $U = \sigma_I \otimes \sigma_I$ and $[\begin{smallmatrix}0 & 0 \\ 0 & \mathcal{I}_1\end{smallmatrix}]$ for $U = \sigma_X \otimes \sigma_I$. This indicates that queries involving $U = \sigma_I \otimes \sigma_I$ are informative about the Hamiltonian parameters $(\omega_0, \delta_0, \phi_0)$ and those involving $U = \sigma_X \otimes \sigma_I$ are informative about $(\omega_1, \delta_1, \phi_1)$.

b. In presence of noise sources and nonidealities

In Sec. IV C 1, we modeled the effect of different noise sources and nonidealities on the quantum system. In particular, we discussed the effect of imperfections in control in Sec. IV C 2, effect of decoherence in Sec. IV C 3 and how the observed measured outcome is subject to readout noise in Sec. IV C 1. We consider the two-qubit decoherence model from Sec. IV C 3, and the readout noise model of a bit-flip channel based on binary classification from Sec. IV C 1. We denote the noisy observed measurement outcome as $\tilde{y}$ and the hidden measurement outcome before the effect of the bit-flip channel as y.

(a) Likelihood. The likelihood function in the presence of the noise sources of readout noise, imperfect-pulse shaping, and decoherence, is then given by

$$p_{\tilde{\mathrm{y}}|\mathrm{x}}(\tilde{y}|x;\boldsymbol{\theta}) = p_{\tilde{\mathrm{y}}|\mathrm{y}}(\tilde{y}|y) \sum_{z \in \{0,1\}} \left[(1 - p_d(t))p_{\mathrm{yz|x}}(yz|(M, U, t + \Delta t_{\mathrm{eff}}(\boldsymbol{\theta}));\boldsymbol{\theta}) + \frac{1}{4}p_d(t)\right] \tag{A18}$$

$$= (1 - p_d(t))[(1 - r_{1-\tilde{y}})p_{\mathrm{y|x}}(\tilde{y}|(M, U, t + \Delta t_{\mathrm{eff}}(\boldsymbol{\theta}));\boldsymbol{\theta}) + r_{\tilde{y}}p_{\mathrm{y|x}}(1 - \tilde{y}|(M, U, t + \Delta t_{\mathrm{eff}}(\boldsymbol{\theta}));\boldsymbol{\theta})] + \frac{1}{2}p_d(t)(1 - r_{1-\tilde{y}} + r_{\tilde{y}}) \tag{A19}$$

with the probability of the two-qubit string yz given by

$$p_{\mathrm{yz|x}}(yz|(M, U, t);\boldsymbol{\theta}) = |\langle yz|Me^{-iH(\boldsymbol{\theta})t}U|00\rangle|^2 \tag{A20}$$

and the probability $p_{\mathrm{y|x}}(y|x)$ given by Eq. (1) (the noiseless case). In the expressions above, we have used the tuple representation of the query x, $p_d(t)$, which is the depolarization probability associated with the two-qubit decoherence model as discussed in Sec. IV C 3 and (r_0, r_1), which are the readout noise parameters as introduced in Sec. IV C 1.

(b) Fisher information matrix. The FI matrix for a given query x is given by

$$\mathcal{I}_x(\boldsymbol{\theta})[i, j] = \frac{1}{p_{\tilde{\mathrm{y}}|\mathrm{x}}(0|x;\boldsymbol{\theta})(1 - p_{\tilde{\mathrm{y}}|\mathrm{x}}(0|x;\boldsymbol{\theta}))} \frac{\partial p_{\tilde{\mathrm{y}}|\mathrm{x}}(0|x;\boldsymbol{\theta})}{\partial \theta_i} \frac{p_{\tilde{\mathrm{y}}|\mathrm{x}}(0|x;\boldsymbol{\theta})}{\partial \theta_j} \tag{A21}$$

where

$$\frac{\partial p_{\tilde{y}|x}(0|x;\boldsymbol{\theta})}{\partial \theta_i} = (1-r_0-r_1)(1-p_d(t))\frac{\partial p_{y|x}(0|x;\boldsymbol{\theta})}{\partial \theta_i} \tag{A22}$$

and

$$\frac{\partial p_{y|x}(0|x;\boldsymbol{\theta})}{\partial \theta_i} = \begin{cases} \frac{\partial p(y=0|(M,U,t+\Delta t_{\text{eff}}(\boldsymbol{\theta}));\boldsymbol{\theta})}{\partial \theta_i}, & \theta_i \notin \{\omega_0,\omega_1\} \\ \frac{\partial p(y=0|(M,U,t+\Delta t_{\text{eff}}(\boldsymbol{\theta}));\boldsymbol{\theta})}{\partial(\theta_i \Delta t_{\text{eff}}(\boldsymbol{\theta}))}\frac{\partial(\theta_i \Delta t_{\text{eff}}(\boldsymbol{\theta}))}{\partial \theta_i}, & \theta_i \in \{\omega_0,\omega_1\} \end{cases}. \tag{A23}$$

Note that special attention has to be given when taking the derivative with respect to $\omega_{0,1}$ as $\Delta t_{\text{eff}}(\boldsymbol{\theta})$ (Sec. IV C 2) actually only has a dependence on these two components in $\boldsymbol{\Lambda}$ and appears in the effective evolution time t with a prefactor of $\omega_{0,1}$ in the likelihood [see Eq. (A13)].

APPENDIX B: COMPUTATIONAL DETAILS OF QUERY OPTIMIZATION

In Sec. V A 1, we pointed out that the number of shots available for each query in the experimental datasets collected from the IBM quantum devices are limited. In this Appendix, we describe how the query optimizations for HAL-FI [Eq. (19)] and HAL-FIR [Eq. (21)] are solved under shot constraints for each query. We then describe how the computational cost of query optimization can be reduced through uncertainty filtering of the query space.

1. Different query optimizations and strategies for handling query constraints

We describe how constraints can be handled for the query optimization [Eq. (19)] in HAL-FI (Algorithm 2) but the same approach can also be used for HAL-FIR. Let us consider the ith round in active learning. The Hamiltonian parameter estimate in this round is $\hat{\boldsymbol{\theta}}^{(i)}$. Let us denote the number of shots available for each query $x \in \mathcal{Q}$ in the ith round as $N^{(i)}_{\text{shots}}(x)$. The number of shots available before learning has started is then $N^{(0)}_{\text{shots}}(x)$. Let the total number of shots that have already been made against query x by the ith round (inclusive) be denoted as $N^{(i)}_{\text{tot}}(x)$. We will denote the total number of shots made over all queries inputted to the oracle by the ith round (inclusive) as $N^{(i)}_{\text{tot}}$.

We can frame the query optimization problem undershot constraints in two different ways, motivated by the asymptotic optimal query distribution associated with HAL-FI,

$$q^{\star} = \arg \min_{q \in \mathcal{P}(\mathcal{Q})} \operatorname{Tr}\big(\mathcal{I}_q^{-1}(\boldsymbol{\theta}^{\star})\big) \tag{B1}$$

where $\mathcal{P}(\mathcal{Q})$ is the family of all probability distributions over the specified query space $\mathcal{Q}$. As discussed in Sec. II C, we cannot solve for this distribution in practice as this requires us to have access to the true parameters $\boldsymbol{\theta}^{\star}$, and hence we solve for a suboptimal query distribution $q^{(i)}(\hat{\boldsymbol{\theta}}^{(i)})$ given the current parameter estimates $\hat{\boldsymbol{\theta}}$. During active learning, we can solve for $q^{(i)}(\hat{\boldsymbol{\theta}}^{(i)})$ by viewing it as the query distribution over all the queries that have inputted to the oracle (i.e., including from previous rounds) or as the query distribution associated with the current ith batch of queries that will be issued. These two viewpoints lead us to two different approaches of how the shot constraints are handled and how the queries to be inputted to the oracle are sampled from the query distribution.

Firstly, consider the following query optimization problem:

$$q^{(i)} = \arg \min_{q} \operatorname{Tr}\big(\mathcal{I}_q^{-1}(\hat{\boldsymbol{\theta}}^{(i)})\big) \tag{B2}$$

$$\text{subject to } \sum_{x \in \mathcal{Q}^{(i)}} q(x) = 1, \text{ and}$$

$$\frac{N^{(i-1)}_{\text{tot}}(x)}{N^{(i)}_{\text{tot}}} \leqslant q(x) \leqslant \min\left\{1, \frac{N^{(i)}_{\text{shots}}(x)}{N^{(i)}_{\text{tot}}}\right\} \forall x \in \mathcal{Q}^{(i)} \tag{B3}$$

with corresponding sampling of queries in that batch as

$$\text{Sample } N_b \text{ queries } X^{(i)}_q \text{ from } \mathcal{Q}^{(i)} \text{ w.p. } q^{(i)}_b \tag{B4}$$

where $q^{(i)}_b(x) = q^{(i)}(x) - \frac{N^{(i-1)}_{\text{tot}}(x)}{N^{(i)}_{\text{tot}}}$. The query optimization in this case considers all the queries that have been sampled from earlier batches during active learning. The query distribution $q^{(i)}$ is then the distribution over all the queries made so far and $q^{(i)}_b$ is the query distribution for the batch to be issued. Further, the Fisher information matrix associated with $q^{(i)}$ is guaranteed to be invertible but that associated with $q^{(i)}_b$ may be noninvertible. This suggests that the outcomes of queries made in this round of the active learning procedure may not be informative about all the Hamiltonian parameters.

In order to ensure that the query distributions for each batch are informative about all the Hamiltonian parameters, one may alternately solve the following query optimization problem:

$$q^{(i)} = \arg \min_{q} \operatorname{Tr}\big(\mathcal{I}_q^{-1}(\hat{\boldsymbol{\theta}}^{(i)})\big) \tag{B5}$$

$$\text{subject to } \sum_{x \in \mathcal{Q}^{(i)}} q(x) = 1, \text{ and}$$

$$0 \leqslant q(x) \leqslant \min\left\{\frac{N^{(i)}_{\text{shots}}(x)}{N^{(i)}_b}, 1\right\}, \forall x \in \mathcal{Q}^{(i)} \tag{B6}$$

where $N^{(i)}_b$ is the size of the batch of queries being issued to the oracle in the ith round. Queries of the issued batch are then sampled as

$$\text{Sample } N_b \text{ queries } X^{(i)}_q \text{ from } \mathcal{Q}^{(i)} \text{ w.p. } q^{(i)} \tag{B7}$$

where $q^{(i)}$ is now the query distribution for each batch. This query optimization can be viewed as a greedy approach of query selection. The Fisher information matrix associated with $q^{(i)}$ is guaranteed to be invertible and hence informative about all the Hamiltonian parameters. We thus solve the query optimization considering shot constraints on each query according to Eq. (B6) for our application.

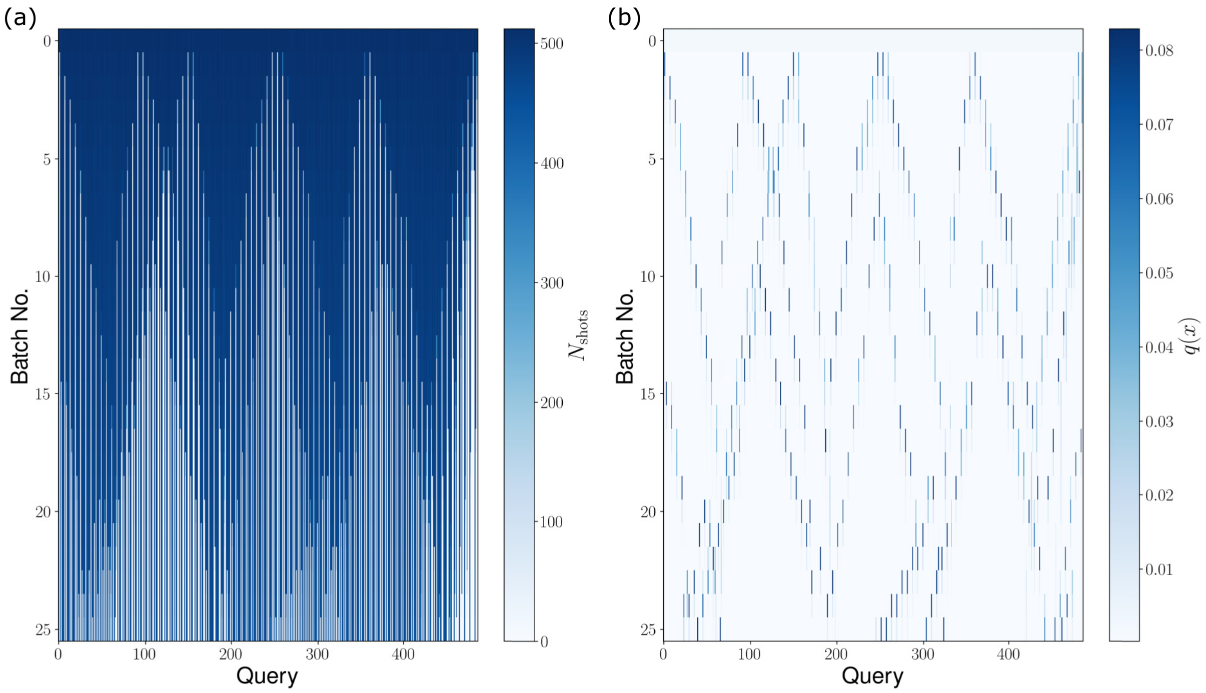

FIG. 19. Visualization of queries being selected for each batch during a particular run of HAL-FI for $N_0 = 2000$ and 25 batches of size $N_b = 5000$. In (a), we plot the available number of shots for each query after batches of queries are made during active learning. In (b), we plot the query distribution for each batch during active learning. The number of shots available for each query before learning starts is $N^{(0)}_{\rm shots}(x) = 512\,\forall x$. Half the total number of shots available in this dataset is exhausted by the end of learning. Note that the different parameter values considered for HAL-FI are for stress testing the query constraints' handling procedure and are not tuned for the HAL-FI algorithm.

As we sample queries randomly according to $q^{(i)}$ and not proportional to $q^{(i)}$, the resulting X_q might not satisfy query constraints exactly and an additional pruning step is required. Moreover, to reduce the computational cost of the query optimization solve, we consider a subset of queries $\mathcal{Q}^{(i)}_{\rm filtered} \subset \mathcal{Q}^{(i)}$ for which shots are still available.

In Algorithm 5, we summarize the steps taken to handle query constraints during query optimization for HAL-FI. It takes the inputs of the query space $\mathcal{Q}^{(i)}$ of the current ith round of active learning, number of shots available for each query and the size of the batch of queries to be issued. The inputted query space is first filtered by retaining only those queries for which shots are available. The resulting filtered query space is denoted by $\mathcal{Q}^{(i)}_{\rm filtered}$, and we then compute the query distribution $q^{(i)}$ according to Eq. (B6). We then sample queries for the batch X_q according to this distribution after mixing and with incorporation of a pruning step. Note that lines 4 and 5 were also used in the original query optimization algorithms of HAL-FI (Algorithm 2) and HAL-FIR (Algorithm 3) to encourage exploration. An illustration of the operation of the query optimization and handling of query constraints is shown in Fig. 19 for a particular run of the HAL-FI learner.

2. Uncertainty filtering of query space

Uncertainty filtering becomes a crucial step during query optimization when we consider HAL-FI with an adaptively growing query space. We noted in Sec. II C 1 d that the computational cost of query optimization [Eq. (19)] scales as $\mathcal{O}(n_{\mathcal{Q}}^2 m^3 + n_{\mathcal{Q}} m^4 + m^5)$ where $n_{\mathcal{Q}} = |\mathcal{Q}|$ is the number of queries in the query space, and m is the length of the parameter vector $\boldsymbol{\theta}$. This computational cost is alleviated through filtering of the query space based on entropy $S(x)$. The entropy of the different queries can be computed from the model probability expressions available to us (see Appendix A) given the

Algorithm 5. Handling query constraints during query optimization in HAL-FI.

```
Input: Current query space $\mathcal{Q}^{(i)}$, number of shots available for each query $N^{(i)}_{\rm shots}(x)$, size of batch of queries requested $N^{(i)}_b$, total number of queries made so far $N^{(i-1)}_{\rm tot}$
Output: Query set $X_q$ of size $N^{(i)}_b$
1: Filter $\mathcal{Q}^{(i)}$ by keeping only queries for which shots are available: $\mathcal{Q}^{(i)}_{\rm filtered} = \{x \in \mathcal{Q}^{(i)} | N^{(i)}_{\rm shots}(x) > 0\}$
2: Obtain HAL-FI query distribution $q^{(i)}$ by solving the query optimization of Eq. (B6) with input of $\mathcal{Q}^{(i)}_{\rm filtered}$
3: Obtain uniform distribution over filtered query space: $p_U = 1/|\mathcal{Q}^{(i)}_{\rm filtered}|$
4: Set mixing coefficient: $\mu = 1 - 1/|N^{(i-1)}_{\rm tot}|^{1/6}$
5: Modify query distribution: $q^{(i)} = \mu q^{(i)} + (1-\mu) p_U$
6: Sample $X_q$ from $\mathcal{Q}^{(i)}_{\rm filtered}$ according to $q^{(i)}$
7: Prune queries from $X_q$ that cannot be made and randomly assign valid queries
8: return $X_q$
```

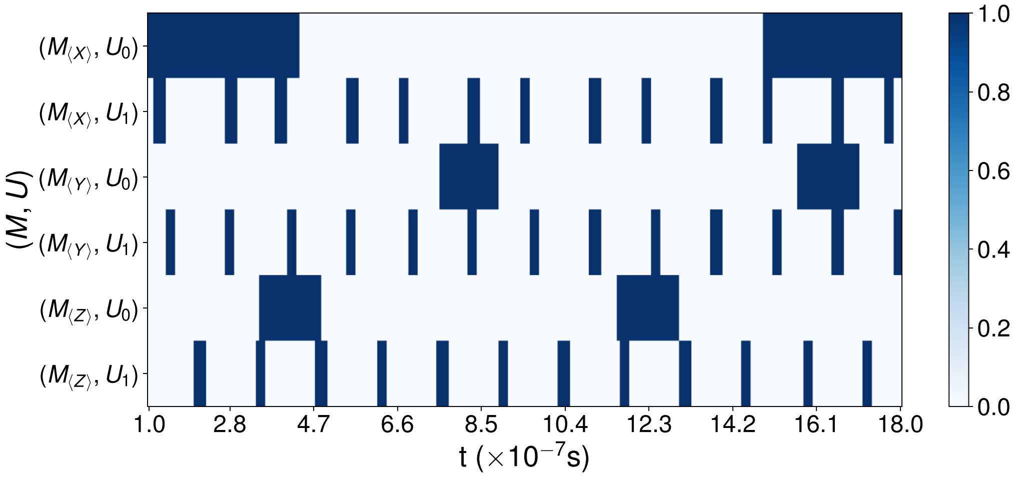

FIG. 20. Uncertainty filtering of $\mathcal{Q}$ as defined in Sec. IV B 2 with $\mathcal{T}$ set to be a sequence of 243 linearly equispaced evolution times in $[10^{-7}, 18 \times 10^{-7}]$ s . The The x axis corresponds to the system evolution times $t \in \mathcal{T}$. The y axis indicates the different combinations of measurement operators and preparation operators available for each query in $\mathcal{Q}$. The different preparation operators are denoted as $U_0 = \sigma_I\sigma_I$ and $U_1 = \sigma_X\sigma_I$. We consider the $\boldsymbol{\theta}^\star$ corresponding to IBM quantum device D *ibmq_boeblingen* under drive configuration 2, with the different noise sources of readout noise, imperfect pulse-shaping, and decoherence being accounted for. Queries colored as dark blue are retained in the filtered query space and rest is filtered out.

current Hamiltonian parameter estimate $\hat{\boldsymbol{\theta}}$. The filtered query space based on entropy is determined as follows:

$$\mathcal{Q}_S = \left\{x | x \in \mathcal{Q}, S(x) > \tau \times \max_{x' \in \mathcal{Q}} S(x')\right\} \quad \text{(B8)}$$

where we set the threshold $\tau = 0.95$, i.e., we only consider queries with entropy that is at least 0.95 times the highest entropy. This value of τ was chosen to ensure that at most only half of the queries are retained after uncertainty filtering of the query space. An illustration of uncertainty filtering of a query space is given in Fig. 20.

APPENDIX C: ESTIMATION PROCEDURE FOR LEARNING CROSS-RESONANCE HAMILTONIANS

In this Appendix, we discuss in detail the estimation procedures of Sec. IV D for the estimator based on regression (Sec. IV D 1) and the MLE estimator (Sec. IV D 2), used for learning cross-resonance Hamiltonians [Eq. (31)]. These estimators can be used by both the passive learner and the active learners of HAL-FI/HAL-FIR (Sec. III). Additionally, we can also use Bayesian estimators such as the sequential Monte Carlo method discussed in Sec. II A 4 c. In the following discussion, we consider the experimental setup from Sec. IV. We will also discuss how these estimation procedures can be improved or extended to other Hamiltonians.

Let us recall the notation introduced in Sec. III and Sec. IV D. The different rounds of active learning are indexed by $i \in [i_{\max}]$. In each round of active learning, we use an estimation procedure divided into multiple steps. We index each of these fractional steps by k. The Hamiltonian parameter estimate at the kth fractional step in the ith round of learning will then be denoted by $\hat{\boldsymbol{\theta}}^{(i,k)}$ with the parameter estimate in the ith round at the end of the estimation procedure denoted simply by $\hat{\boldsymbol{\theta}}^{(i)}$. The training examples available at the ith round is given by $(X^{(i)}, Y^{(i)})$.

We now discuss the estimation procedures for (i) regression and (ii) maximum-likelihood estimation (MLE).

1. Regression

As mentioned in Sec. IV D 1, the first step of our estimation procedure involves frequency estimation followed by a nonlinear regression solve combined with a gradient descent procedure on the Rabi oscillations inferred from data [Eq. (39)]. We denoted these inferred Rabi oscillations by $\hat{p}_{\text{rabi}}(t)$.

Estimates are obtained by fitting nonlinear regression equations of the form $A\cos(\omega t) + B\sin(\omega t) + C$ to the Rabi oscillations, where the fit minimizes the sum of the weighted L_2 errors across the corresponding time series. For a given value of ω, the coefficients A, B, and C are estimated using weighted linear least-squares regression. Fast Fourier transforms (FFTs) [92] are used to perform an efficient grid search over these weighted least-squares fits to obtain initial estimates of ω. Bracketed gradient-based search is then used to refine the estimates. The resulting coefficients A, B, and C for each Rabi oscillation are subsequently used to obtain estimates of the corresponding δ and ϕ Hamiltonian parameters. We note this analysis in general assumes periodic signals. In order to successfully estimate the Rabi frequencies using standard FFT, the set of time points $\mathcal{T}$ should span at least the time duration established by the minimum-frequency criterion defined in Sec. V C.

For each Rabi oscillation, the following weighted least-squares *normal* equations are solved to estimate the A, B, C coefficients for a given value of ω, where the summations are performed over time points $t \in \mathcal{T}$, and where the w_t's are weighting factors

$$\begin{bmatrix} \sum_t w_t \cos^2(\omega t) & \sum_t w_t \cos(\omega t)\sin(\omega t) & \sum_t w_t \cos(\omega t) \\ \sum_t w_t \cos(\omega t)\sin(\omega t) & \sum_t w_t \sin^2(\omega t) & \sum_t w_t \sin(\omega t) \\ \sum_t w_t \cos(\omega t) & \sum_t w_t \sin(\omega t)\sin(\omega t) & \sum_t w_t \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_t w_t \hat{p}_{\text{rabi}}(t)\cos(\omega t) \\ \sum_t w_t \hat{p}_{\text{rabi}}(t)\sin(\omega t) \\ \sum_t w_t \hat{p}_{\text{rabi}}(t) \end{bmatrix}. \tag{C1}$$

The weights w_t are equal to one over the estimation variance of $\hat{p}_{\text{rabi}}(t)$ for the corresponding time points t. The purpose of the weights is to account for the heteroskedasticity of the estimation errors in $\hat{p}_{\text{rabi}}(t)$.

If we write Eq. (C1) in the form $R\mathbf{a} = \mathbf{b}$, where $\mathbf{a}$ is the vector of A, B, C coefficients that satisfies Eq. (C1), then the residual weighted squared error is given by

$$\mathcal{E}^2 = \sum_t w_t \hat{p}^2_{\text{rabi}}(t) - \mathbf{a}^T R\mathbf{a}. \tag{C2}$$

For a group of Rabi oscillations that share the same frequency ω, the optimization problem is to find the ω that minimizes the sum of the weighted squared errors $\mathcal{E}^2$ across those time series. Fourier techniques are used to make the initial search computationally efficient by noting that Eq. (C1) can be rewritten as

$$\begin{bmatrix} \frac{1}{2}\sum_t w_t + \frac{1}{2}\sum_t w_t \cos(2\omega t) & \frac{1}{2}\sum_t w_t \sin(2\omega t) & \sum_t w_t \cos(\omega t) \\ \frac{1}{2}\sum_t w_t \sin(2\omega t) & \frac{1}{2}\sum_t w_t - \frac{1}{2}\sum_t w_t \cos(2\omega t) & \sum_t w_t \sin(\omega t) \\ \sum_t w_t \cos(\omega t) & \sum_t w_t \sin(\omega t) & \sum_t w_t \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_t w_t \hat{p}_{\text{rabi}}(t)\cos(\omega t) \\ \sum_t w_t \hat{p}_{\text{rabi}}(t)\sin(\omega t) \\ \sum_t w_t \hat{p}_{\text{rabi}}(t) \end{bmatrix}, \tag{C3}$$

which simplifies to

$$\begin{bmatrix} \frac{1}{2}\mathcal{F}\{w_t\}(0) + \frac{1}{2}\text{Re}(\mathcal{F}\{w_t\}(2\omega)) & -\frac{1}{2}\text{Im}(\mathcal{F}\{w_t\}(2\omega)) & \text{Re}(\mathcal{F}\{w_t\}(\omega)) \\ -\frac{1}{2}\text{Im}(\mathcal{F}\{w_t\}(2\omega)) & \frac{1}{2}\mathcal{F}\{w_t\}(0) - \frac{1}{2}\text{Re}(\mathcal{F}\{w_t\}(2\omega)) & -\text{Im}(\mathcal{F}\{w_t\}(\omega)) \\ \text{Re}(\mathcal{F}\{w_t\}(\omega)) & -\text{Im}(\mathcal{F}\{w_t\}(\omega)) & \mathcal{F}\{w_t\}(0) \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \text{Re}(\mathcal{F}\{w_t \hat{p}_{\text{rabi}}(t)\}(\omega)) \\ -\text{Im}(\mathcal{F}\{w_t \hat{p}_{\text{rabi}}(t)\}(\omega)) \\ \mathcal{F}\{w_t \hat{p}_{\text{rabi}}(t)\}(0) \end{bmatrix} \tag{C4}$$

where

$$\mathcal{F}\{x(t)\}(\omega) = \sum_t x(t)e^{-i\omega t} \tag{C5}$$

is the discrete-time Fourier transform of $x(t)$ over $t \in \mathcal{T}$, $x(t) \in \{w_t, w_t\hat{p}_{\text{rabi}}(t)\}$. Standard FFTs allow us to efficiently calculate the various Fourier coefficients in Eq. (C4) at fixed intervals in the spectrum using

$$F\left\{x\left(\frac{n}{F_s} + t_0\right)\right\}(k) = \sum_{n=0}^{N-1} x\left(\frac{n}{F_s} + t_0\right)e^{-i\frac{2\pi}{N}kn}. \tag{C6}$$

The Fourier coefficients calculated via the above equation can then be used in Eq. (C4) to perform a computationally efficient grid search over possible values of $\omega \in \{1\frac{2\pi F_s}{N}, 2\frac{2\pi F_s}{N}, 3\frac{2\pi F_s}{N}, \ldots, \pi F_s\}$, with an initial estimate for ω obtained by minimizing Eq. (C2) summed over the corresponding Rabi oscillations. Bracketed gradient-based search can then be performed using Eq. (C1) directly at a higher computational cost to refine these grid-search estimates.

For grid search purposes, we have also found it useful to sample frequency amplitudes at intermediate half steps to better avoid local minima, particularly when $\mathcal{T}$ does not satisfy the minimum-frequency criteria. Such intermediate sampling is accomplished by multiplying signals of the form $x(\frac{n}{F_s} + t_0)$ by a rotating exponential frequency and then computing a second FFT,

$$\begin{aligned} G\left\{x\left(\frac{n}{F_s} + t_0\right)\right\}(k) &= F\left\{x\left(\frac{n}{F_s} + t_0\right)\right\}\left(k + \frac{1}{2}\right) \\ &= \sum_{n=0}^{N-1}\left[x\left(\frac{n}{F_s} + t_0\right)e^{-i\frac{\pi}{N}n}\right]e^{-i\frac{2\pi}{N}kn}. \end{aligned} \tag{C7}$$

For general multiparameter Hamiltonians, the above initial estimation procedure must be modified to ensure that all the frequency components in a Rabi oscillation corresponding to (M, U) can be faithfully extracted. The model must then assume that each Fourier series has up to K modes. Normal equations for the same can then be setup. Such an approach has been employed in Bayesian spectral analysis [93] and nonstationary time-series estimation [94,95].

2. Maximum likelihood estimation

The estimation procedure for MLE (Sec. IV D 2) requires an initial estimate $\hat{\boldsymbol{\theta}}^{(i,0)}$, which ideally lives in the same convex basin as the global minimum of the MLE. This allows for a more localized search to be carried out and a stochastic gradient descent procedure should allow us to jump out of any smaller local minima here if present. When an initial estimate cannot be provided by the learner, the estimates obtained through regression as just discussed is used as an initial guess to the MLE.

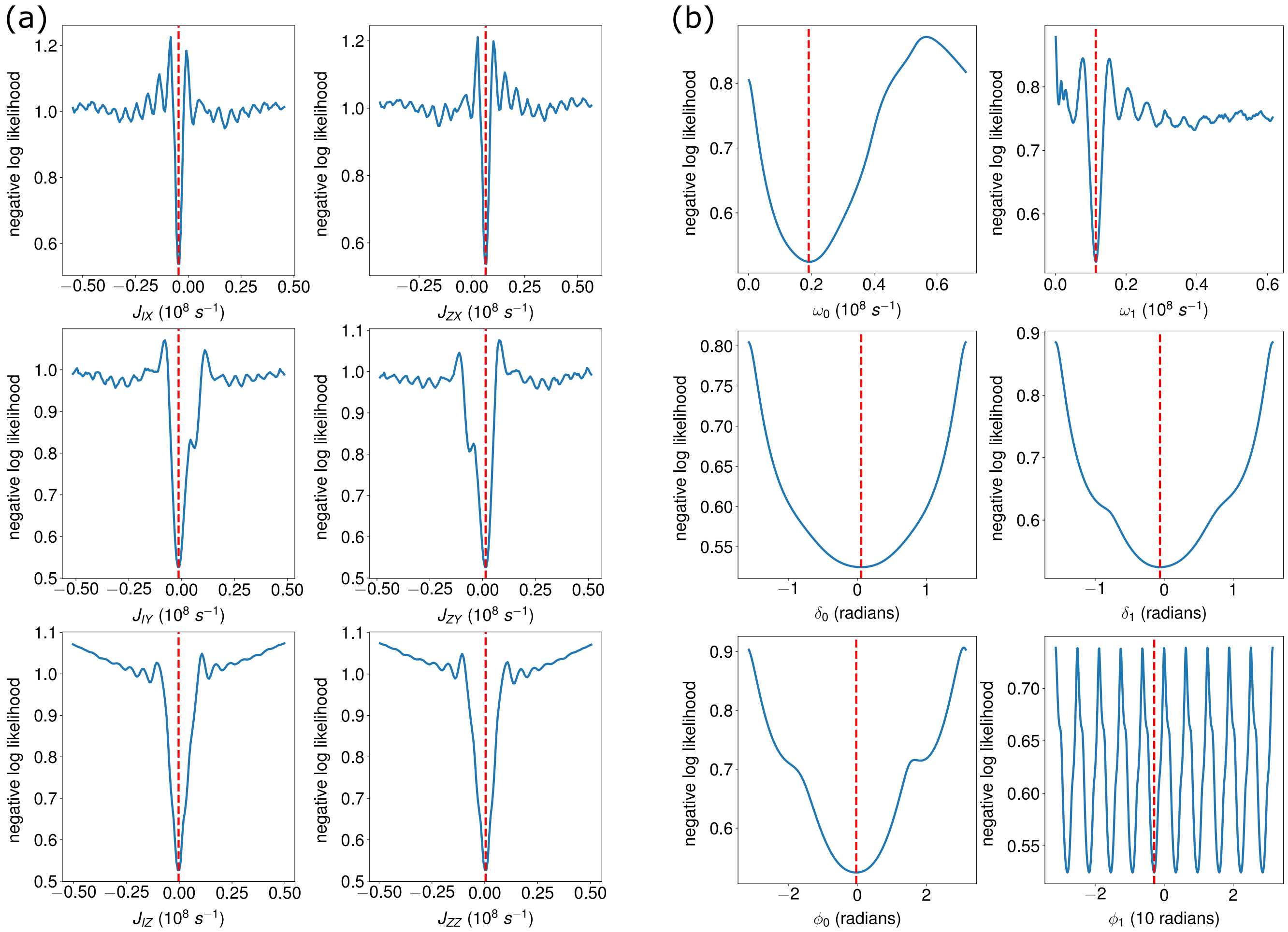

FIG. 21. Slices of energy landscapes of the log-likelihood loss function along the different parameter components considering experimental data collected from IBM quantum device D *ibmq_boeblingen* under drive configuration 3 (Table IV) (a) using parametrization $\mathbf{J}$, and (b) using parametrization $\mathbf{\Lambda}$. In each slice of θ_i (x axis), we fix the values of the other components to that obtained through estimation and evaluate the negative log-likelihood loss (y axis) by changing the value of θ_i. We indicate the Hamiltonian parameter estimate $\hat{\theta}_i$ as obtained through our estimation procedure by a dashed red line.

We solve the MLE 5 through a combination of SGD applied on different parametrizations and the quasi-Newton method for further refinement. Addition of the latter step helps us in saving computationally expensive hyperparameter tuning that is required.

(1) SGD solve using ADAM considering the $\mathbf{\Lambda}$ parametrization and learning rate of $\eta^i_{\mathbf{\Lambda}}$ using the input of $\hat{\boldsymbol{\theta}}^{(i,0)}$. This returns the output of $\hat{\boldsymbol{\theta}}^{(i,1)}$.

(2) SGD solve using ADAM considering the $\mathbf{J}$ parametrization and learning rate of $\eta^i_{\mathbf{J}}$ using the input of $\hat{\boldsymbol{\theta}}^{(i,1)}$. This returns the output of $\hat{\boldsymbol{\theta}}^{(i,2)}$.

(3) MLE solve considering the $\mathbf{J}$ parametrization using LFBGS-B using the input of $\hat{\boldsymbol{\theta}}^{(i,2)}$. This returns the output of $\hat{\boldsymbol{\theta}}^{(i,3)}$.

We set the learning rate η^i for ADAM [79] in the ith round of active learning according to the number of queries already made. We consider the learning rate to be $\eta^i \propto \frac{1}{\sqrt{|X^{(i)}|}}$, i.e., the learning rate is reduced inversely to the square root of the number of training examples. This ensures a more localized search as we progress in the learning. We consider $\eta^0 = 10^{-3}$. We found that carrying out step 2 after step 1 gave us more accurate estimates of $\hat{\theta}$ than just carrying out step 1. Moreover, after the first few rounds of HAL, we can skip steps 1 and 2. We can carry out step 3 directly using an initial condition of $\hat{\theta}^{(i,0)}$ from initial estimation or $\hat{\theta}^{(i-1)}$ from the previous round.

3. Energy landscapes of negative log-likelihood loss for cross-resonance Hamiltonian

We ascertain the efficacy of the estimation procedure by visualizing the energy landscapes of the negative log-likelihood loss function [Eq. (56)] and inspecting the location of the $\hat{\boldsymbol{\theta}}$ in the landscape. In Fig. 21, we plot the energy landscape obtained from an experimental dataset, for the two different parametrizations $\mathbf{J}$ and $\mathbf{\Lambda}$. The energy landscapes indicate the nonlinear and non-convex nature of the MLE of Eq. (56). These energy landscapes also indicate why solving the MLE in the parametrization of $\mathbf{\Lambda}$ using ADAM is carried out before

TABLE VIII. Summary of estimated CR Hamiltonian parameters for the IBM quantum devices A, B, and C. We give the Hamiltonian parameters in the parametrization $\mathbf{J}$ and the physically relevant frequency components in $\mathbf{\Lambda}$. The readout noise is defined by the parameters of r_0 and r_1, which are the conditional probabilities of bit flip given the measurement outcomes are $y = 0$ and $y = 1$ respectively (see Sec. IV C 1). We show the results for CR Hamiltonians between two different pairs of qubits on Device C, specified as (control qubit, target qubit): (0,1) (CR_{01}) and (0,2) (CR_{02}).

Device	Drive config.	Hamiltonian parameters [$\times 10^6\, s^{-1}$]		Noise sources	
		$\mathbf{J} = (J_{IX}, J_{IY}, J_{IZ}, J_{ZX}, J_{ZY}, J_{ZZ})$	(ω_0, ω_1)	Readout (r_0, r_1)	Time offset $(\Delta t_{\mathrm{eff},0}, \Delta t_{\mathrm{eff},1})$ [ns]
A	0	(58.47, 3.68, −5.00, 10.76, 2.29, −0.52)	(69.71, 47.94)	(0.160,0.215)	(15, 101)
	1	(38.93, 2.34, 0.26, 11.15, −3.30, −0.30)	(50.09, 28.36)	(0.150, 0.210)	(97, 187)
	2	(19.35, 0.12, 0.69, 10.80, −0.66, 0.24)	(30.17, 8.60)	(0.220,0.150)	(178, 699)
	3	(−0.21, −1.68, 0.20, 10.47, 1.50, −0.86)	(10.28, 11.20)	(0.145, 0.150)	(579, 532)
	4	(−20.11, −1.35, 0.73, 10.55, 0.94, −1.13)	(9.58, 30.80)	(0.190, 0.185)	(594, 174)
B	0	(30.03, 3.62, 0.49, 1.75, −0.16, −0.31)	(31.97, 28.54)	(0.110, 0.140)	(149, 175)
	1	(15.34, 1.85, 0.19, 1.81, −0.75, −0.59)	(17.19, 13.80)	(0.090, 0.070)	(318, 407)
	2	(0.89, 0.72, 0.24, 1.82, −0.54, −0.34)	(2.72, 1.67)	(0.120, 0.160)	(2235, 3712)
	3	(−13.71, −2.31, −0.54, 1.78, 0.09, 0.09)	(12.15, 15.69)	(0.130, 0.160)	(472, 353)
	4	(−28.45, −2.19, −1.20, 1.56, 2.15, 0.27)	(26.91, 30.35)	(0.110, 0.100)	(187, 161)
$\mathrm{C/CR}_{01}$	0	(−8.52, −2.15, −0.26, 10.93, 0.85, 0.32)	(2.74, 19.69)	(0.200, 0.160)	(2280, 286)
	1	(−3.88, −2.26, −0.35, 10.88, 1.46, 0.43)	(7.04, 15.24)	(0.120, 0.160)	(869, 376)
	2	(0.58, −1.81, −0.45, 10.81, 0.83, 1.24)	(11.46, 10.70)	(0.080, 0.070)	(518, 563)
	3	(4.86, −1.66, 0.08, 10.85, 0.44, −0.14)	(15.75, 6.35)	(0.070, 0.110)	(369, 966)
	4	(9.53, −0.17, 0.29, 10.76, −0.17, −0.32)	(20.29, 1.36)	(0.070, 0.120)	(276, 4835)
$\mathrm{C/CR}_{02}$	0	(9.42, −0.71, 0.27, 12.21, −0.71, −0.25)	(21.67, 2.84)	(0.070, 0.060)	(255, 2123)
	1	(6.17, −0.46, 0.09, 11.96, −0.59, −0.26)	(18.16, 5.81)	(0.070, 0.110)	(311, 1038)
	2	(2.59, 0.05, −0.26, 11.90, −1.66, −0.17)	(14.58, 9.46)	(0.050, 0.090)	(397, 629)
	3	(−1.03, −0.10, −0.16, 11.99, −0.63, −0.18)	(10.99, 13.03)	(0.090, 0.060)	(541, 445)
	4	(−4.53, 0.10, −0.31, 12.04, −0.18, 0.39)	(7.50, 16.59)	(0.100, 0.110)	(811, 344)

solving the MLE in the parametrization of $\mathbf{J}$. The slices along specific components of $\mathbf{\Lambda}$ display more convex like nature than those along components of $\mathbf{J}$. It should also be noted that there is a global minimum present in the energy landscapes, which we are able to identify using our estimation procedure.

4. Incorporating uncertainty from shot noise

The inferred Rabi oscillations $p_{\mathrm{rabi}}(t)$ used for estimation are sensitive to the number of shots made for a given query $x = (M, U, t)$. This variability in the Rabi oscillations leads to a variability in the estimates of $\hat{\boldsymbol{\theta}}^{(i,0)}$ produced. This variability is particularly high during the initial rounds of HAL when there are only a few shots of each query present in the set of training examples. In order to accurately account for this variability and hence include the uncertainty in our estimates of the Hamiltonian parameters, we consider the following procedure. Let us consider the initial estimation of Algorithm 4 as the procedure applied on a particular realization of the Rabi oscillations. We construct n_{rep} realizations of the Rabi oscillations considering the observed Rabi oscillation data and sampling according to the binomial distribution associated with the number of shots for each query. For each of these realizations, we obtain frequency estimates of $\omega_{0,1}$ through the initial estimation as described above. We then fit the parametric distributions of log-normal distributions to frequency estimates from each each realization. We then continue with the initial estimation procedure and MLE considering each of these realizations.

Other ways of incorporating uncertainty in the Hamiltonian parameter estimates during the estimation procedure would be to adopt a Bayesian learning framework or stochastic process regression (e.g., Gaussian process regression). This is left for future work.

APPENDIX D: LEARNED HAMILTONIAN PARAMETERS AND LEARNING ERROR ON IBM QUANTUM DEVICES

In this Appendix, we give a summary of the estimated cross-resonance Hamiltonian parameters on different IBM quantum devices not already discussed in Sec. V and lend further support to the performance of different learners on the 20-qubit IBM quantum device D *ibmq_boeblingen* under drive configuration 2 (Table V A 2).

1. Summary of model parameters on IBM quantum devices

In Sec. V A 2, we described the estimated Hamiltonian parameters and the noise model parameters for the IBM quantum device D *ibmq_boeblingen*. Here, we give a similar summary for the other IBM quantum devices (A, B, and C).

Considering the entire experimental datasets collected for each of these IBM quantum devices (Sec. IV B 1) as training data, we compute the Hamiltonian parameters using our estimation procedure (Sec. IV D), and that of the different noise sources (Sec. IV C). We summarize these parameters for the different devices in Table VIII. These Hamiltonian parameters serve as approximations of the true parameters $\mathbf{J}^{\star}$. Here, the

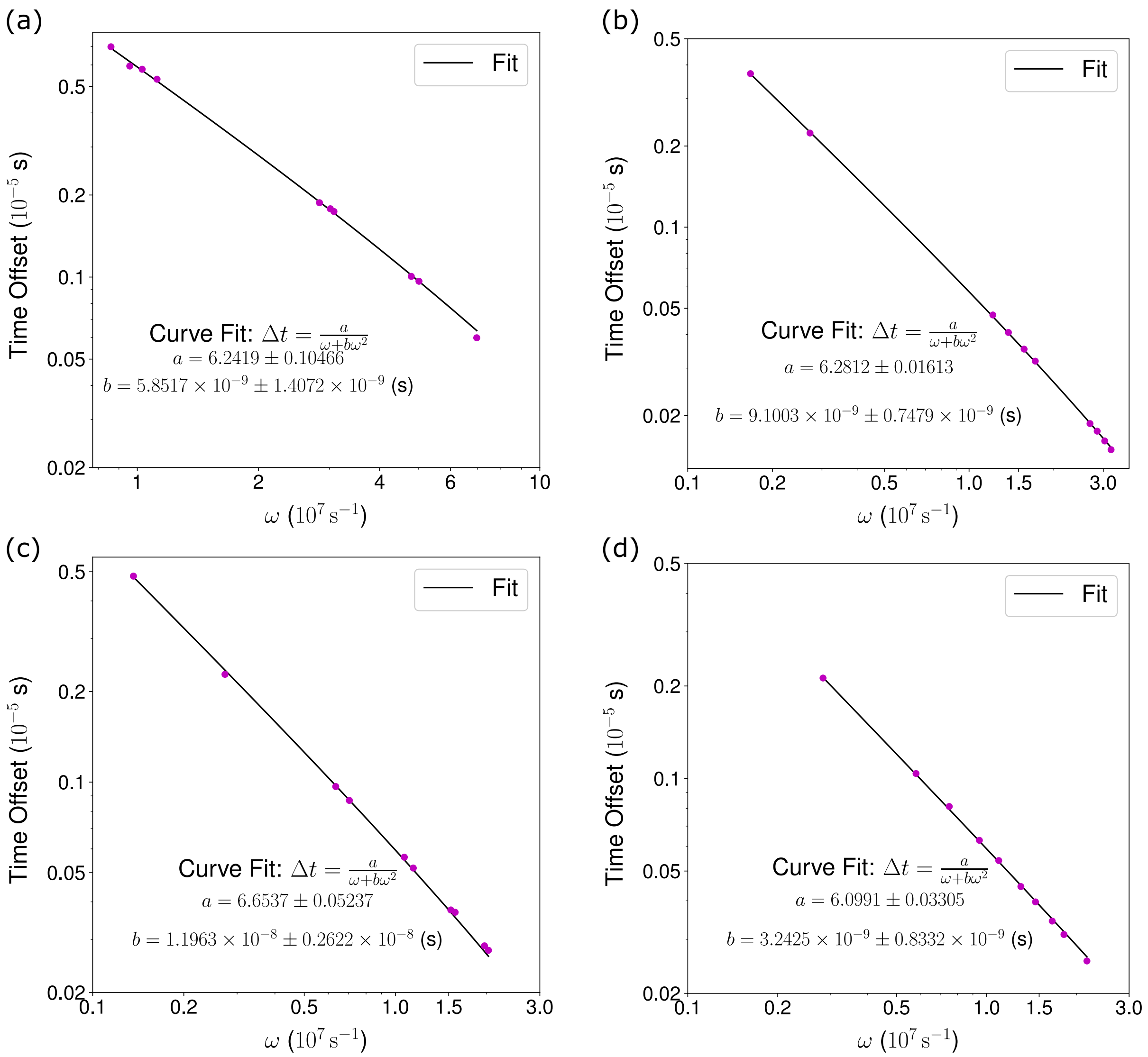

FIG. 22. Dependence of the time offset Δt on parameters ω for IBM quantum devices (a) A, (b) B, (c) C CR$_{01}$, and (d) C CR$_{02}$. The plotted data points correspond to driving the device under different conditions and hence different cross-resonance Hamiltonians. The imperfect pulse shaping model extracted from these experimental data points is shown by a fit and this is later used in the MLE.

readout noise parameters (r_0, r_1) are given to indicate the amount of readout noise possible in these devices and serve as a proxy for the conditional distributions of the readout given the measurement outcome used in the final MLE [Eq. (56)]. The time offset $\Delta t_{\mathrm{eff},i}$ introduced due to imperfect control is also specified and where the subscript i indicates dependence on the preparation operator $U_0 = \sigma_I \sigma_I$ and $U_1 = \sigma_X \sigma_I$.

As mentioned earlier in Sec. IV C 2, we fit the estimated values of Δt_{eff} to the Hamiltonian parameters to obtain a model for the time-offset. This is shown in Fig. 22 and is used in the MLE [Eq. (56)].

2. Expected trends of learning error

In Sec. V C, we assessed the performance of the HAL-FI and HAL-FIR algorithms in different learning scenarios on IBM quantum device D *ibmq_boeblingen* under drive configuration 2, where the query distribution was learned in real time. Here, we lend support that the trends observed in Figs. 10–12 are expected.

To determine the behavior of the learners in an idealized setting, we consider the case where we have access to the optimal query distribution during learning. In Fig. 23, we show the trend of RMSE for HAL-FI with a fixed query space assuming access to the query distribution $q(\boldsymbol{\theta}^\star)$ during training and that the Cramer-Rao bound is saturated. We follow the same protocol from Sec. V A 5 as we did for our earlier experiments. For the baseline strategy and a passive learner, the query distribution corresponds to an uniform distribution over the query space. As expected, using a passive learner does not change the scaling in the finite query nor the asymptotic query regimes. A scaling of $\epsilon \sim 1/\sqrt{N}$ or $N \sim \epsilon^{-2}$ is observed, which is in line with the standard quantum limit (SQL). For HAL-FI, we observe an initial scaling in RMSE with number of queries, which is higher than SQL but this reduces to SQL in the asymptotic query regime. Thus, our

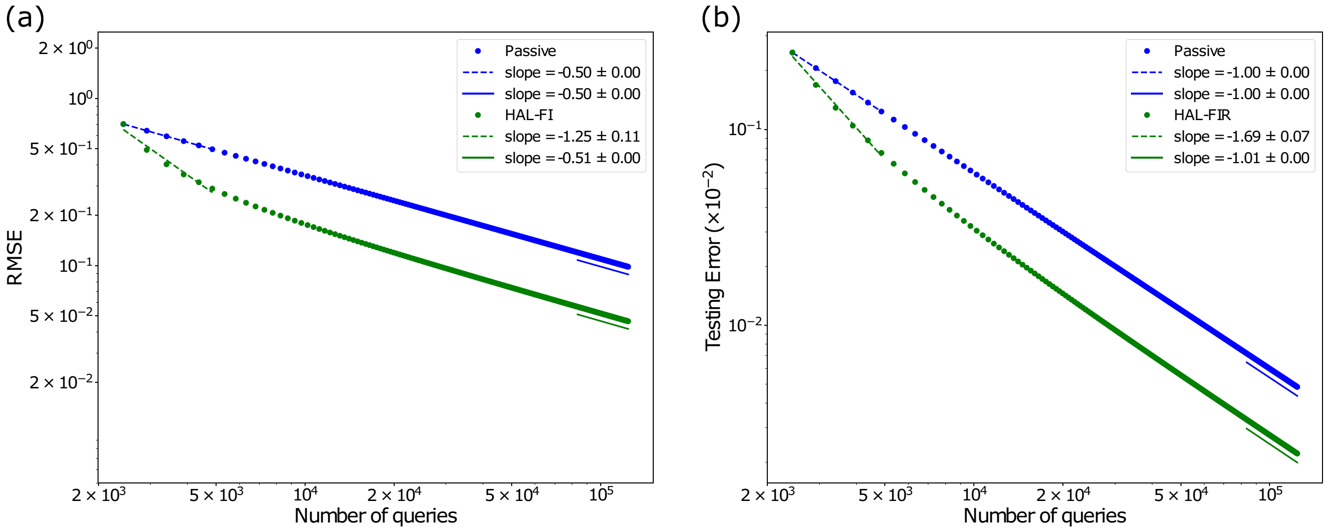

FIG. 23. Scaling of learning testing error with number of queries for Hamiltonian learning considering access to the asymptotic optimal query distribution $q(\boldsymbol{\theta}^\star)$ of HAL-FI or HAL-FIR. We show trends of (a) RMSE and (b) testing error obtained upon analysis of the lower Cramer-Rao bounds. We consider the Hamiltonian parameters of $\boldsymbol{\theta}^\star$ as determined from the experimental dataset of IBM quantum device D *ibmq_boeblingen* under drive configuration 2. We plot the trends of learning error with number of queries for HAL-FI/HAL-FIR against the passive learner, which uses the uniform distribution over $\mathcal{Q}$.

results from Sec. V C are in agreement with what we observe here. Asymptotically, we expect a constant savings in the number of queries or resources required when employing an active learner with a fixed query space compared to a passive learner.

Likewise for Hamiltonian learning with prior information, we can compute the RMSE with number of queries for HAL-FI with an adaptively growing query space and access to $q^{(i)}(\boldsymbol{\theta}^\star)$ for each ith batch during learning. In Fig. 24, we show the trend of RMSE with number of queries for HAL-FI with a linearly growing query space and an exponentially growing query space. We observe that the passive learner has a scaling of the SQL in the asymptotic query regime. HAL-FI in the linearly growing query space achieves super-Heisenberg scaling. We note that this supports the trend of RMSE achieved during the experiments in Sec. V D 1.

As noted in Sec. V D 1, HAL-FI with an exponentially growing query space also achieves Heisenberg limited scaling until the evolution times being included in the query space reach the magnitude of the decoherence time T_1 and T_2. In this case, HAL-FI avoids selecting higher evolution times as the information gained from these measurement outcomes will tend to zero. Thus, we expect that the Heisenberg limit to be achieved for the finite query setting.

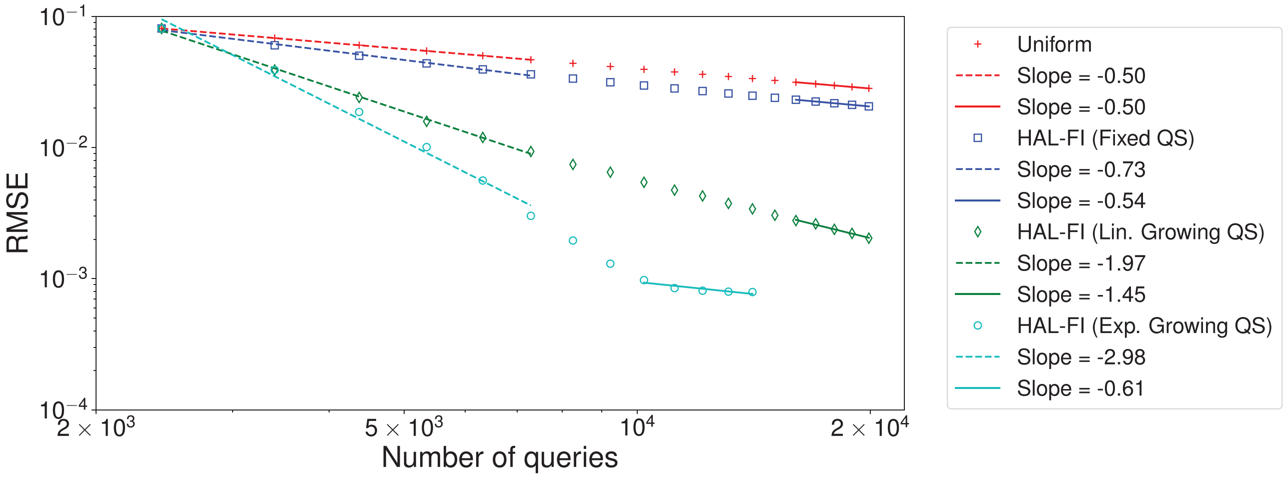

FIG. 24. Hamiltonian learning with access to prior information of subset of parameters during recalibration: Scaling of RMSE with Number of Queries. We assume access to the asymptotic optimal query distribution $q(\boldsymbol{\theta}^\star)$ of HAL-FI and analysis of the Cramer-Rao bounds. We consider the Hamiltonian parameters of $\boldsymbol{\theta}^\star$ as determined from the experimental dataset of IBM quantum device D *ibmq_boeblingen* under drive configuration 3.

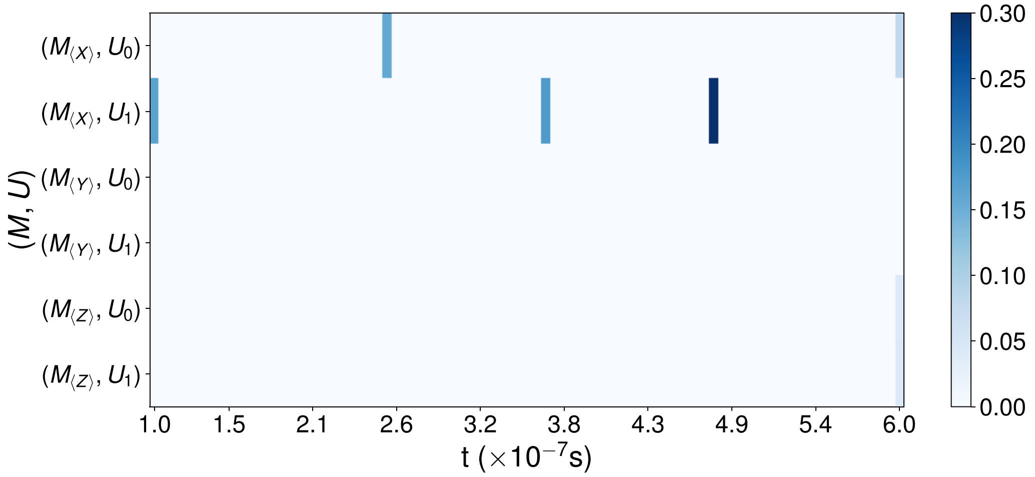

FIG. 25. Asymptotic optimal query distribution $q(\boldsymbol{\theta}^\star)$ for HAL-FI with a fixed query space (Sec. IV B 2) on *ibmq_boeblingen* under drive configuration 2 (Table IV). We consider different noise sources of readout noise, imperfect pulse shaping, and decoherence. The y axis indicates the different combinations of measurement operators and preparation operators available for each query in $\mathcal{Q}$. The different preparation operators are denoted as $U_0 = \sigma_I\sigma_I$ and $U_1 = \sigma_X\sigma_I$. The x axis corresponds to $\mathcal{T}$, which is set to 81 equispaced evolution times in $[10^{-7}, 6 \times 10^{-7}]$ s . The query distribution is colorcoded according to the colormap on the right.

3. Sparse query distributions

Another consequence of using HAL-FI is the sparsity of the asymptotic query distribution during learning. This is confirmed by visualizing the optimal query distribution of HAL-FI with the fixed query space (Sec. IV B 2) considering IBM quantum device *ibmq_boeblingen* under drive configuration 2 in Fig. 25. It is interesting to note that this was achieved even though sparsity was not incorporated into the learning problem. This can be explained by realizing that the most informative queries are in fact sparse over the query space. It should be noted, however, that this is typically not the query distribution that HAL-FI has access to during learning as the true parameters $\boldsymbol{\theta}^\star$ are not available and the query distribution obtained through optimization [Eq. (19)] is modified by mixing with the uniform distribution (see Sec. III A).

APPENDIX E: HEISENBERG LIMITED SCALING IN CROSS-RESONANCE TYPE HAMILTONIANS

In this Appendix, we discuss if Heisenberg limited scaling (HLS) can be achieved in different quantum systems with simplified cross-resonance Hamiltonians. The simplified cross-resonance Hamiltonians are obtained by removing particular Pauli product terms from the cross-resonance (CR) Hamiltonian (Sec. IV A) that we have studied so far. We give examples of such Hamiltonians (equipped with the query space described in Sec. IV) in Appendix E 1 where HLS is achieved for all the parameters and examples where HLS is not achieved for all the parameters during Hamiltonian learning in Appendix E 2. For examples of HLS, we describe query distributions obtained by HAL-FI and relate them to query distributions obtained through zero crossings of Rabi oscillations or maximum entropy.

In the following examples, we consider reduction to a single interaction in the two qubit system and a three interaction example, which we call the simplified cross-resonance (SCR) Hamiltonian. This reduction from the full CR Hamiltonian is achieved by setting the appropriate Hamiltonian parameters to zero and introducing the matrix R that denotes which parts of the CR Fisher information matrix are involved in the estimation of these respective Hamiltonian parameters. Suppose the reduced set of parameters are collected into the vector $\boldsymbol{\theta}_R$, then the Cramer-Rao bound is now

$$\sum_i \mathrm{Var}((\boldsymbol{\theta}_R)_i) \geqslant \frac{1}{N}\mathrm{Tr}\big(\mathcal{I}_q^{-1}(\boldsymbol{\theta}_R)\big) = \frac{1}{N}\mathrm{Tr}\big(R^{-T}\mathcal{I}_q^{-1}(\boldsymbol{\theta})R^{-1}\big) \tag{E1}$$

where $\mathcal{I}_q(\boldsymbol{\theta}_R)$ is the reduced Fisher information matrix corresponding to the query distribution q. In the last step, we noted the relation of the reduced Fisher information matrix with the full Fisher information matrix as $\mathcal{I}_q(\boldsymbol{\theta}_R) = R\mathcal{I}_q(\boldsymbol{\theta})R^T$.

1. Examples

a. Single interaction two-qubit system

The Hamiltonian of interest in this case is

$$H = J_{ZX}\sigma_Z \otimes \sigma_X. \tag{E2}$$

This may be obtained by considering the parameter set of $\mathbf{J} = (0, 0, 0, J_{ZX}, 0, 0)^T$ and $R = [0, 0, 0, 1, 0, 0]$. The values of the parameters in the alternate parametrization of $\boldsymbol{\Lambda} = (\omega_0, \delta_0, \phi_0, \omega_1, \delta_1, \phi_1)^T = (J_{ZX}, 0, 0, J_{ZX}, 0, \pi)$ where we have assumed $J_{ZX} > 0$. The Rabi oscillations for different queries in this case are as follows:

$$(M_{\langle X\rangle}, U_j, t) : p_{\mathrm{rabi}}(x) = 0, \tag{E3}$$

$$(M_{\langle Y\rangle}, U_j, t) : p_{\mathrm{rabi}}(x) = \sin(2\omega_j t), \tag{E4}$$

$$(M_{\langle Z\rangle}, U_j, t) : p_{\mathrm{rabi}}(x) = \cos(2\omega_j t), \tag{E5}$$

where the index $j \in \{0, 1\}$ corresponds to different preparation operators $U_0 = \sigma_I\sigma_I$ and $U_1 = \sigma_X\sigma_I$. We note that the measurement operator of $M_{\langle X\rangle}$ is not informative about the frequency $\omega_0 = J_{ZX}$ for this system and can also be noted from considering the corresponding Fisher information. In order to learn the parameter of interest ω_0, it is enough to consider one of the queries in $\{M_{\langle Y\rangle}, M_{\langle Z\rangle}\} \times \{\sigma_I\sigma_I, \sigma_X\sigma_I\}$ and a suitable time range $\mathcal{T}$.

Let us select the query of $(M_{\langle Z\rangle}, \sigma_I\sigma_I, t)$ and suppose our query distribution over the time range is based on the zeros the Rabi oscillations. Given a time range $\mathcal{T}$, we consider values of $t_k = \frac{\pi}{4\omega_0} + \frac{k\pi}{2\omega_0}$ where $k \in \mathbb{N}$. Queries with these system evolution times have the maximum entropy for the considered (M, U).

Note that the Fisher information of a query in this case is given by $\mathcal{I}_x(\omega_0) = 4t^2$. Through the Cramer-Rao bound, we then have

$$\epsilon^2 = \text{Var}(\omega_0) \geqslant \sum_{k=1}^{N} \frac{1}{4t_k^2}$$
$$\approx \begin{cases} \frac{1}{2}\frac{\omega^2}{N\pi^2}, & \text{fixed space of zero crossings} \\ \frac{3}{4}\frac{\omega^2}{N^3\pi^2}, & \text{linearly spaced zero crossings} \end{cases} \tag{E6}$$

where we have set the learning error to be achieved as ϵ. We can thus expect to achieve a scaling of $N \sim \epsilon^{-3/2}$ when using linearly spaced zero crossings. For exponentially spaced zero crossings of the Rabi oscillations, the variance approaches zero at an increasing rate.

b. Two interaction systems

Consider the following Hamiltonian:

$$H = J_{IX}\sigma_I \otimes \sigma_X + J_{ZX}\sigma_Z \otimes \sigma_X. \tag{E7}$$

The reduced set of parameters is then $\mathbf{\Lambda}_R = (\omega_0, \omega_1) = (|J_{IX} + J_{ZX}|, |J_{IX} - J_{ZX}|)$. As in the earlier single interaction example, we can choose the queries that contain the zero crossings of the Rabi oscillations. The Rabi oscillations are given by

$$(M_{\langle X\rangle}, U_j, t) : p_{\text{rabi}}(x) = 0, \tag{E8}$$

$$(M_{\langle Y\rangle}, U_j, t) : p_{\text{rabi}}(x) = \sin(2\omega_j t), \tag{E9}$$

$$(M_{\langle Z\rangle}, U_j, t) : p_{\text{rabi}}(x) = \cos(2\omega_j t), \tag{E10}$$

where $j \in \{0, 1\}$ is used as an index to denote the preparation operators $U_0 = \sigma_I\sigma_I$ and $U_1 = \sigma_X\sigma_I$. A complete set of queries to estimate (ω_0, ω_1) with Heisenberg limited scaling would then be

$$\mathcal{Q} = \left\{(M_{\langle Z\rangle}, \sigma_I\sigma_I, t_k) : t_k = \frac{\pi}{4\omega_0} + \frac{k\pi}{2\omega_0}, k \in \mathbb{N}\right\}$$
$$\bigcup \left\{(M_{\langle Z\rangle}, \sigma_X\sigma_I, t_k) : t_k = \frac{\pi}{4\omega_1} + \frac{k\pi}{2\omega_1}, k \in \mathbb{N}\right\}. \tag{E11}$$

The set of evolution times chosen here also correspond to those with maximum entropy. The Fisher information of a query made through either set of measurement or preparation operators in the query space is given by $\mathcal{I}_x(\omega_0, \omega_1) = 4t^2$. Thus, through the argument we made in the previous section, we can also learn the parameters of unitary here with Heisenberg limited scaling.

2. Examples of non-HLS scaling during Hamiltonian learning

So far, we have given examples of Hamiltonians obtained through simplification of the CR Hamiltonian, that can be learned with HLS scaling. We now give examples of Hamiltonians, which cannot be learned with HLS scaling using the query space described in Sec. IV B 2.

a. Two interaction systems

Now, let us consider the following alternate Hamiltonian (modified slightly from the previous example discussed):

$$H = J_{IY}\sigma_I \otimes \sigma_Y + J_{ZX}\sigma_Z \otimes \sigma_X \tag{E12}$$

where a complete reduced set of parameters is $\mathbf{\Lambda_R} = (\omega_0, \phi_0) = (\sqrt{J_{ZX}^2 + J_{IY}^2}, \tan^{-1}(\frac{J_{IY}}{J_{ZX}}))$. This set of parameters contains a frequency in addition to a phase. Fisher information matrices in the $\mathbf{\Lambda}_R$ parametrization considering queries of the form $(M, \sigma_I\sigma_I, t)$ where we select one particular preparation operator is given by

$$M_{\langle X\rangle} : \mathcal{I} = \frac{1}{1 - \sin^2(\phi_0)\sin^2(2\omega_0 t)} \begin{bmatrix} 4t^2\sin^2(\phi_0)\cos^2(2\omega_0 t) & \frac{1}{2}t\sin(2\phi_0)\sin(4\omega_0 t) \\ \frac{1}{2}t\sin(2\phi_0)\sin(4\omega_0 t) & \cos^2(\phi_0)\sin^2(2\omega_0 t) \end{bmatrix}, \tag{E13}$$

$$M_{\langle Y\rangle} : \mathcal{I} = \frac{1}{1 - \cos^2(\phi_0)\sin^2(2\omega_0 t)} \begin{bmatrix} 4t^2\cos^2(\phi_0)\cos^2(2\omega_0 t) & -\frac{1}{2}t\sin(2\phi_0)\sin(4\omega_0 t) \\ -\frac{1}{2}t\sin(2\phi_0)\sin(4\omega_0 t) & \sin^2(\phi_0)\sin^2(2\omega_0 t) \end{bmatrix}, \tag{E14}$$

$$M_{\langle Z\rangle} : \mathcal{I} = \begin{bmatrix} 4t^2 & 0 \\ 0 & 0 \end{bmatrix}. \tag{E15}$$

Fisher information matrices in $\mathbf{J}_R$ is given by

$$\mathcal{I}_J = \begin{bmatrix} \frac{J_{IY}^2}{\omega_0^2}\mathcal{I}_{11} + \frac{2J_{IY}J_{ZX}}{\omega_0^3}\mathcal{I}_{12} + \frac{J_{ZX}^2}{\omega_0^4}\mathcal{I}_{22} & \frac{J_{ZX}}{\omega_0^3}\mathcal{I}_{12} - \frac{J_{IY}}{\omega_0^3}\mathcal{I}_{12} + \frac{J_{IY}J_{ZX}}{\omega_0^2}(\mathcal{I}_{11} - \mathcal{I}_{12}) \\ . & \frac{J_{ZX}^2}{\omega_0^2}\mathcal{I}_{11} - \frac{2J_{IY}J_{ZX}}{\omega_0^3}\mathcal{I}_{12} + \frac{J_{IY}^2}{\omega_0^4}\mathcal{I}_{22} \end{bmatrix} \tag{E16}$$

where we have related them to elements of $\mathcal{I}$ as given above in Eq. (E15). We observe that in order to obtain HLS in (J_{IY}, J_{ZX}), it is necessary to set $\mathcal{I}_{22} = 0$ and $\mathcal{I}_{12} \neq 0$ to ensure $\mathcal{I}_J$ is full rank and there is an explicit dependence on the variable t that we can take advantage of, for HLS. However, the required conditions cannot be achieved simultaneously here. This suggests that the current set of queries cannot be used to achieve HLS.

b. Three interaction simplified cross-resonance gate

The Hamiltonian of interest in this case is

$$H = J_{IX}\sigma_I \otimes \sigma_X + J_{IY}\sigma_I \otimes \sigma_Y + J_{ZX}\sigma_Z \otimes \sigma_X. \tag{E17}$$

This may be obtained by considering the parameter set of $\mathbf{J} = (J_{IX}, J_{IY}, 0, J_{ZX}, 0, 0)^T$. Defining R is a bit more tricky in this case compared to the previous example. Let us first look at the alternate parametrization of $\mathbf{\Lambda} = (\sqrt{(J_{IX}+J_{ZX})^2 + J_{IY}^2}, 0, \tan^{-1}(\frac{J_{IY}}{J_{IX}+J_{ZX}}), \sqrt{(J_{IX} - J_{ZX})^2+J_{IY}^2}, 0, \tan^{-1}(\frac{J_{IY}}{J_{IX}-J_{ZX}}))^T$. Note that the reduced parametrization of Λ_R is an over-parameterization with four non-zero components compared to $\mathbf{J}_R$, which only has three nonzero components. The Rabi oscillations in this case are given by

$$(M_{\langle X\rangle}, U_j, t):\ p_{\text{rabi}}(x) = \sin(\phi_j)\sin(2\omega_j t), \tag{E18}$$

$$(M_{\langle Y\rangle}, U_j, t):\ p_{\text{rabi}}(x) = \cos(\phi_j)\sin(2\omega_j t), \tag{E19}$$

$$(M_{\langle Z\rangle}, U_j, t):\ p_{\text{rabi}}(x) = \cos(2\omega_j t). \tag{E20}$$

The Fisher information matrices can be obtained by looking at the CR Fisher information matrices (Appendix A) and simplifying them,

$$M_{\langle X\rangle} : \mathcal{I}_j = \frac{1}{1-\sin^2(\phi_j)\sin^2(2\omega_j t)}\begin{bmatrix} 4t^2\sin^2(\phi_j)\cos^2(2\omega_j t) & \frac{1}{2}t\sin(2\phi_j)\sin(4\omega_j t) \\ \frac{1}{2}t\sin(2\phi_j)\sin(4\omega_j t) & \cos^2(\phi_j)\sin^2(2\omega_j t)\end{bmatrix}, \tag{E21}$$

$$M_{\langle Y\rangle} : \mathcal{I}_j = \frac{1}{1-\cos^2(\phi_j)\sin^2(2\omega_j t)}\begin{bmatrix} 4t^2\cos^2(\phi_j)\cos^2(2\omega_j t) & -\frac{1}{2}t\sin(2\phi_j)\sin(4\omega_j t) \\ -\frac{1}{2}t\sin(2\phi_j)\sin(4\omega_j t) & \sin^2(\phi_j)\sin^2(2\omega_j t)\end{bmatrix}, \tag{E22}$$

$$M_{\langle Z\rangle} : \mathcal{I}_j = \begin{bmatrix} 4t^2 & 0 \\ 0 & 0\end{bmatrix}. \tag{E23}$$

If we were to consider the zero crossings of the Rabi oscillations as in the previous example, the queries and their corresponding Fisher information matrices are of the following form:

$$M_{\langle X\rangle} : t_k(M_{\langle X\rangle}) = \frac{\pi}{2\omega_j} + \frac{k\pi}{2\omega_j},\ \mathcal{I}_j = \begin{bmatrix} 4t_k^2\sin^2(\phi_j) & 0 \\ 0 & 0\end{bmatrix}, \tag{E24}$$

$$M_{\langle Y\rangle} : t_k(M_{\langle Y\rangle}) = \frac{\pi}{2\omega_j} + \frac{k\pi}{2\omega_j},\ \mathcal{I}_j = \begin{bmatrix} 4t_k^2\cos^2(\phi_j) & 0 \\ 0 & 0\end{bmatrix}, \tag{E25}$$

$$M_{\langle Z\rangle} : t_k(M_{\langle Z\rangle}) = \frac{\pi}{4\omega_j} + \frac{k\pi}{2\omega_j},\ \mathcal{I}_j = \begin{bmatrix} 4t_k^2 & 0 \\ 0 & 0\end{bmatrix}, \tag{E26}$$

where $k \in \mathbb{N}$. It should be noted that the evolution times $t_k(M)$ being selected are a function of the measurement operator involved in the query, which is made explicit through the argument M. As $\mathbf{\Lambda}_R$ is an overparameterization, let us look at the query Fisher information matrix $\mathcal{I}_q(\mathbf{J}_R)$ for the above set of queries,

$$\mathcal{I}_q(\mathbf{J}_R) = \sum_k \sum_{M\in\{M_{\langle X\rangle}, M_{\langle Z\rangle}\}} 4t_k^2(M)\begin{bmatrix} \left(\frac{\partial\omega_0}{\partial J_{IX}}\right)^2 + \left(\frac{\partial\omega_1}{\partial J_{IX}}\right)^2 & \left(\frac{\partial\omega_0}{\partial J_{IX}}\frac{\partial\omega_0}{\partial J_{IY}} + \frac{\partial\omega_1}{\partial J_{IX}}\frac{\partial\omega_1}{\partial J_{IY}}\right) & \left(\frac{\partial\omega_0}{\partial J_{IX}}\frac{\partial\omega_0}{\partial J_{ZX}} + \frac{\partial\omega_1}{\partial J_{IX}}\frac{\partial\omega_1}{\partial J_{ZX}}\right) \\ & \left(\frac{\partial\omega_0}{\partial J_{IY}}\right)^2 + \left(\frac{\partial\omega_1}{\partial J_{IY}}\right)^2 & \left(\frac{\partial\omega_0}{\partial J_{IY}}\frac{\partial\omega_0}{\partial J_{ZX}} + \frac{\partial\omega_1}{\partial J_{IY}}\frac{\partial\omega_1}{\partial J_{ZX}}\right) \\ & & \left(\frac{\partial\omega_0}{\partial J_{ZX}}\right)^2 + \left(\frac{\partial\omega_1}{\partial J_{ZX}}\right)^2 \end{bmatrix} \tag{E27}$$

where we have only given the upper-triangular part of the symmetric matrix. It can be shown that for these queries, $\mathcal{I}_q(\mathbf{J}_R)$ is rank deficient and thus noninvertible. This was foreshadowed by the fact that $\mathcal{I}_q(\mathbf{\Lambda})$ was informative in ω_0 and ω_1 but not one of the phases $\phi_{0,1}$. Hence, it is more appropriate to consider $R = [\begin{smallmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0\end{smallmatrix}]$ for these set of queries. It can be verified through an analysis similar to the single interaction system that we can achieve a scaling of $N \sim O(\epsilon^{-3/2})$ and hence make improvements over than SQL.

If we wish to learn J_{IY} as well, it is necessary to introduce other queries such that the Fisher information matrix is non-zero for the corresponding parameter of interest. Let us start by changing our learning task to the simpler challenge of learning the parameters (ω_0, δ_0). In this case, it is enough to consider only queries of the form $(M_{\langle X\rangle}, \sigma_I \otimes \sigma_I, t)$ where the time range $t \in \mathcal{T}$

needs to be determined. We immediately observe that

$$\mathcal{I}_q(\mathbf{\Lambda}_R)^{-1} \propto \sum_k \frac{1}{1-\sin^2(\phi_0)\sin^2(2\omega_0 t_k)} \begin{bmatrix} \cos^2(\phi_j)\sin^2(2\omega_j t) & -\frac{1}{2}t\sin(2\phi_j)\sin(4\omega_j t) \\ -\frac{1}{2}t\sin(2\phi_j)\sin(4\omega_j t) & 4t_k^2\sin^2(\phi_j)\cos^2(2\omega_j t) \end{bmatrix} \tag{E28}$$

and the variance of parameter ϕ_0,

$$\mathrm{Var}(\phi_0) \geqslant \frac{1}{N}\sum_{k=1}^{N} \frac{1-\sin^2(\phi_0)\sin^2(2\omega_0 t_k)}{\cos^2(\phi_0)\sin^2(2\omega_0 t_k)} \tag{E29}$$

where the term inside the sum on the right-hand side is fixed for any periodic or equispaced set of evolution times t_k and thus HLS cannot be achieved using such a set of queries. One key to ensure achieving Heisenberg limited scaling is to introduce an explicit dependence on the variable of system evolution time t into the corresponding Fisher information. We note that this is not followed by the different set of measurement operators considered here.

[1] J. H. Cole, S. G. Schirmer, A. D. Greentree, C. J. Wellard, D. K. L. Oi, and L. C. L. Hollenberg, Identifying an experimental two-state Hamiltonian to arbitrary accuracy, Phys. Rev. A **71**, 062312 (2005).

[2] J. H. Cole, A. D. Greentree, D. K. L. Oi, S. G. Schirmer, C. J. Wellard, and L. C. L. Hollenberg, Identifying a two-state Hamiltonian in the presence of decoherence, Phys. Rev. A **73**, 062333 (2006).

[3] D. Burgarth and K. Yuasa, Quantum System Identification, Phys. Rev. Lett. **108**, 080502 (2012).

[4] A. Valenti, E. van Nieuwenburg, S. Huber, and E. Greplova, Hamiltonian learning for quantum error correction, Phys. Rev. Res. **1**, 033092 (2019).

[5] J. Wang, S. Paesani, R. Santagati, S. Knauer, A. A. Gentile, N. Wiebe, M. Petruzzella, J. L. O'Brien, J. G. Rarity, A. Laing *et al.*, Experimental quantum Hamiltonian learning, Nat. Phys. **13**, 551 (2017).

[6] E. Farhi and S. Gutmann, Analog analogue of a digital quantum computation, Phys. Rev. A **57**, 2403 (1998).

[7] R. Demkowicz-Dobrzański and M. Markiewicz, Quantum computation speedup limits from quantum metrological precision bounds, Phys. Rev. A **91**, 062322 (2015).

[8] N. Liu, J. Thompson, C. Weedbrook, S. Lloyd, V. Vedral, M. Gu, and K. Modi, Power of one qumode for quantum computation, Phys. Rev. A **93**, 052304 (2016).

[9] L. Innocenti, L. Banchi, A. Ferraro, S. Bose, and M. Paternostro, Supervised learning of time-independent Hamiltonians for gate design, New J. Phys. **22**, 065001 (2020).

[10] M. Mohseni, A. T. Rezakhani, and D. A. Lidar, Quantum-process tomography: Resource analysis of different strategies, Phys. Rev. A **77**, 032322 (2008).

[11] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum-enhanced measurements: Beating the standard quantum limit, Science **306**, 1330 (2004).

[12] M. Zwierz, C. A. Pérez-Delgado, and P. Kok, General Optimality of the Heisenberg Limit for Quantum Metrology, Phys. Rev. Lett. **105**, 180402 (2010).

[13] V. Giovannetti, S. Lloyd, and L. Maccone, Advances in quantum metrology, Nat. Photonics **5**, 222 (2011).

[14] G. Tóth and I. Apellaniz, Quantum metrology from a quantum information science perspective, J. Phys. A: Math. Theor. **47**, 424006 (2014).

[15] L. Pezzè, A. Smerzi, M. K. Oberthaler, R. Schmied, and P. Treutlein, Quantum metrology with nonclassical states of atomic ensembles, Rev. Mod. Phys. **90**, 035005 (2018).

[16] H. Imai and A. Fujiwara, Geometry of optimal estimation scheme for *SU(D)* channels, J. Phys. A: Math. Theor. **40**, 4391 (2007).

[17] D. W. Berry, M. Tsang, M. J. W. Hall, and H. M. Wiseman, Quantum Bell-Ziv-Zakai Bounds and Heisenberg Limits for Waveform Estimation, Phys. Rev. X **5**, 031018 (2015).

[18] M. Szczykulska, T. Baumgratz, and A. Datta, Multi-parameter quantum metrology, Adv. Phys.: X **1**, 621 (2016).

[19] W. Górecki, R. Demkowicz-Dobrzański, H. M. Wiseman, and D. W. Berry, π-Corrected Heisenberg Limit, Phys. Rev. Lett. **124**, 030501 (2020).

[20] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Metrology, Phys. Rev. Lett. **96**, 010401 (2006).

[21] J. J. Bollinger, W. M. Itano, D. J. Wineland, and D. J. Heinzen, Optimal frequency measurements with maximally correlated states, Phys. Rev. A **54**, R4649(R) (1996).

[22] H. Lee, P. Kok, and J. P. Dowling, A quantum Rosetta stone for interferometry, J. Mod. Opt. **49**, 2325 (2002).

[23] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information*: 10th Anniversary Edition* (Cambridge University Press, Cambridge, UK, 2010)

[24] B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, and G. J. Pryde, Entanglement-free Heisenberg-limited phase estimation, Nature (London) **450**, 393 (2007).

[25] H. M. Wiseman, D. W. Berry, S. D. Bartlett, B. L. Higgins, and G. J. Pryde, Adaptive measurements in the optical quantum information laboratory, IEEE J. Sel. Top. Quantum Electron. **15**, 1661 (2009).

[26] B. L. Higgins, D. W. Berry, S. D. Bartlett, M. W. Mitchell, H. M. Wiseman, and G. J. Pryde, Demonstrating Heisenberg-limited unambiguous phase estimation without adaptive measurements, New J. Phys. **11**, 073023 (2009).

[27] S. Kimmel, G. H. Low, and T. J. Yoder, Robust calibration of a universal single-qubit gate set via robust phase estimation, Phys. Rev. A **92**, 062315 (2015).

[28] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, A. Bengtsson, S. Boixo, M. Broughton, B. B. Buckley *et al.*, Observation of separated dynamics of charge and spin in the Fermi-Hubbard model, arXiv:2010.07965

[29] T. J. Evans, R. Harper, and S. T. Flammia, Scalable Bayesian Hamiltonian learning, arXiv:1912.07636.

[30] S. Krastanov, S. Zhou, S. T. Flammia, and L. Jiang, Stochastic estimation of dynamical variables, Quantum Sci. Technol. **4**, 035003 (2019).

[31] R. Kosut, I. A. Walmsley, and H. Rabitz, Optimal experiment design for quantum state and process tomography and Hamiltonian parameter estimation, arXiv:quant-ph/0411093.

[32] J. M. Geremia and H. Rabitz, Optimal Identification of Hamiltonian Information by Closed-Loop Laser Control of Quantum Systems, Phys. Rev. Lett. **89**, 263902 (2002).

[33] J. Nunn, B. J. Smith, G. Puentes, I. A. Walmsley, and J. S. Lundeen, Optimal experiment design for quantum state tomography: Fair, precise, and minimal tomography, Phys. Rev. A **81**, 042109 (2010).

[34] Y. Gazit, H. K. Ng, and J. Suzuki, Quantum process tomography via optimal design of experiments, Phys. Rev. A **100**, 012350 (2019).

[35] H. Yuan and C.-H. F. Fung, Optimal Feedback Scheme and Universal Time Scaling for Hamiltonian Parameter Estimation, Phys. Rev. Lett. **115**, 110401 (2015).

[36] N. Kura and M. Ueda, Finite-error metrological bounds on multiparameter Hamiltonian estimation, Phys. Rev. A **97**, 012101 (2018).

[37] C. E. Granade, C. Ferrie, N. Wiebe, and D. G. Cory, Robust online Hamiltonian learning, New J. Phys. **14**, 103013 (2012).

[38] C. Granade, C. Ferrie, I. Hincks, S. Casagrande, T. Alexander, J. Gross, M. Kononenko, and Y. Sanders, Qinfer: Statistical inference software for quantum applications, Quantum **1**, 5 (2017).

[39] J. Sourati, M. Akcakaya, T. K. Leen, D. Erdogmus, and J. G. Dy, Asymptotic analysis of objectives based on Fisher information in active learning, J. Mach. Learn. Res. **18**, 1123 (2017).

[40] A. H. Jazwinski, *Stochastic Processes and Filtering Theory* (Courier Corporation, New York, 2007).

[41] D. Alspach and H. Sorenson, Nonlinear Bayesian estimation using gaussian sum approximations, IEEE Trans. Autom. Control **17**, 439 (1972).

[42] A. Doucet, A. M. Johansen *et al.*, A tutorial on particle filtering and smoothing: Fifteen years later, *Handbook of Nonlinear Filtering* (Oxford University Press, Oxford, 2009), Vol. 12, p. 3.

[43] D. D. Lewis and W. A. Gale, in *SIGIR'94* (Springer, New York, 1994), pp. 3–12.

[44] B. Settles, Active learning literature survey, Tech. Rep. 1648 (University of Wisconsin-Madison, 2009).

[45] H. S. Seung, M. Opper, and H. Sompolinsky, in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT'92 (Association for Computing Machinery, New York, 1992), pp. 287–294.

[46] R. Burbidge, J. J. Rowland, and R. D. King, in *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL'07* (Springer-Verlag, Berlin, Heidelberg, 2007), pp. 209–218.

[47] M.-F. Balcan, A. Broder, and T. Zhang, in *Proceedings of the 20th Annual Conference on Learning Theory, COLT'07* (Springer-Verlag, Berlin, 2007), pp. 35–50.

[48] K. Yu, J. Bi, and V. Tresp, in *Proceedings of the 23rd International Conference on Machine Learning, ICML'06* (Association for Computing Machinery, 2006), pp. 1081–1088.

[49] P. Flaherty, A. Arkin, and M. Jordan, in *Advances in Neural Information Processing Systems*, edited by Y. Weiss, B. Schölkopf, and J. Platt (MIT Press, Cambridge, 2006), Vol. 18, pp. 363–370.

[50] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao, Exploring representativeness and informativeness for active learning, IEEE Trans. Cybern. **47**, 14 (2015).

[51] S.-J. Huang, R. Jin, and Z.-H. Zhou, Active learning by querying informative and representative examples, IEEE Trans. Pattern Anal. Mach. Intell. **36**, 1936 (2014).

[52] B. Settles, Active Learning, *Synthesis Lectures on Artificial Intelligence and Machine Learning* (Morgan & Claypool, New York, 2012), Vol. 6, pp. 1-114.

[53] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley-Interscience, New York, 2006).

[54] E. L. Lehmann and G. Casella, *Theory of Point Estimation* (Springer Science & Business Media, New York, 2006).

[55] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference* (Springer Science & Business Media, New York, 2013).

[56] D. G. Fischer, S. H. Kienle, and M. Freyberger, Quantum-state estimation by self-learning measurements, Phys. Rev. A **61**, 032306 (2000).

[57] F. Huszár and N. M. T. Houlsby, Adaptive Bayesian quantum tomography, Phys. Rev. A **85**, 052120 (2012).

[58] C. Ferrie, C. E. Granade, and D. G. Cory, How to best sample a periodic probability distribution, or on the accuracy of Hamiltonian finding strategies, Quantum Inf. Process. **12**, 611 (2013).

[59] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, in *Proceedings of the 23rd International Conference on Machine Learning, ICML'06* (ACM, New York, 2006), pp. 417–424.

[60] K. Chaudhuri, S. M. Kakade, P. Netrapalli, and S. Sanghavi, in *Advances in Neural Information Processing Systems, NIPS'15* (ACM, New York, 2015), pp. 1090–1098.

[61] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, in *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, STOC'12* (Association for Computing Machinery, New York, 2012), pp. 563–578.

[62] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'12* (SIAM, Philadelphia, 2012), pp. 1183–1194.

[63] T. Zahavy, M. Haroush, N. Merlis, D. J. Mankowitz, and S. Mannor, in *Advances in Neural Information Processing Systems, NIPS'18* (Curran Associates, Red Hook, NY, 2018), pp. 3566–3577.

[64] W. Masson, P. Ranchod, and G. Konidaris, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16* (AAAI Press, Palo Alto, CA, 2016), pp. 1934–1940.

[65] G. Dulac-Arnold, R. Evans, H. van Hasselt, P. Sunehag, T. Lillicrap, J. Hunt, T. Mann, T. Weber, T. Degris, and B. Coppin, Deep reinforcement learning in large discrete action spaces, arXiv:1512.07679.

[66] I. Hincks, T. Alexander, M. Kononenko, B. Soloway, and D. G. Cory, Hamiltonian learning with online Bayesian experiment design in practice, arXiv:1806.02427.

[67] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, 2018).

[68] K. Wei, R. Iyer, and J. Bilmes, in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, edited by F. Bach and D. Blei (PMLR, Lille, France, 2015), Vol. 37, pp. 1954–1963.

[69] C. Rigetti and M. Devoret, Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies, Phys. Rev. B **81**, 134507 (2010).

[70] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, and M. Steffen, Simple All-Microwave Entangling Gate for Fixed-Frequency Superconducting Qubits, Phys. Rev. Lett. **107**, 080502 (2011).

[71] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, Procedure for systematically tuning up cross-talk in the cross-resonance gate, Phys. Rev. A **93**, 060302(R) (2016).

[72] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, Geometric theory of nonlocal two-qubit operations, Phys. Rev. A **67**, 042313 (2003).

[73] E. Magesan and J. M. Gambetta, Effective hamiltonian models of the cross-resonance gate, Phys. Rev. A **101**, 052308 (2020).

[74] J. M. Chow, J. M. Gambetta, E. Magesan, D. W. Abraham, A. W. Cross, B. Johnson, N. A. Masluk, C. A. Ryan, J. A. Smolin, S. J. Srinivasan *et al.*, Implementing a strand of a scalable fault-tolerant quantum computing fabric, Nat. Commun. **5**, 4015 (2014).

[75] T. Alexander, N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, A. Javadi-Abhari, and D. C. McKay, Qiskit Pulse: Programming quantum computers through the cloud with pulses, Quantum Sci. Technol. **5**, 044006 (2020).

[76] H. Abraham, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, E. Arbel, A. Asfaw, C. Azaustre, P. Barkoutsos, G. Barron, L. Bello *et al.*, *Qiskit: An Open-source Framework for Quantum Computing* (Zenodo, 2019).

[77] N. Sundaresan, I. Lauer, E. Pritchett, E. Magesan, P. Jurcevic, and J. M. Gambetta, Reducing unitary and spectator errors in cross resonance with optimized rotary echoes, PRX Quantum **1**, 020318 (2020).

[78] J. Wallman, C. Granade, R. Harper, and S. T. Flammia, Estimating the coherence of noise, New J. Phys. **17**, 113020 (2015).

[79] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[80] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. **23**, 550 (1997).

[81] J. Liu and M. West, in *Sequential Monte Carlo Methods in Practice* (Springer-Verlag, New York, 2001), pp. 197–223.

[82] In [37,38], the package for statistical inference altogether is called Qinfer and the sequential active learner is but just a part of it.

[83] L. Cohen, Time-frequency distributions – A review, Proc. IEEE **77**, 941 (1989).

[84] R. Parhizkar, Y. Barbotin, and M. Vetterli, Sequences with minimal time–frequency uncertainty, Appl. Comput. Harmon. Anal. **38**, 452 (2015).

[85] A. Anshu, S. Arunachalam, T. Kuwahara, and M. Soleimanifar, Sample-efficient learning of interacting quantum systems, Nat. Phys., **17** 931 (2021).

[86] D. M. Abrams, N. Didier, S. A. Caldwell, B. R. Johnson, and C. A. Ryan, Methods for Measuring Magnetic Flux Crosstalk Between Tunable Transmons, Phys. Rev. Appl. **12**, 064022 (2019).

[87] X. Dai, D. M. Tennant, R. Trappen, A. J. Martinez, D. Melanson, M. A. Yurtalan, Y. Tang, S. Novikov, J. A. Grover, S. M. Disseler, J. I. Basham, R. Das, D. K. Kim, A. J. Melville, B. M. Niedzielski, S. J. Weber, J. L. Yoder, D. A. Lidar, and A. Lupascu, Calibration of flux crosstalk in large-scale flux-tunable superconducting quantum circuits, PRX Quantum **2**, 040313 (2021).

[88] K. Pang, M. Dong, Y. Wu, and T. Hospedales, Meta-learning transferable active learning policies by deep reinforcement learning, arXiv:1806.04798 (2018).

[89] A. Dutt, HAL (2023), https://doi.org/10.5281/zenodo.7859448.

[90] A. Dutt, BayesianHL (2023), https://doi.org/10.5281/zenodo.7859452.

[91] D. C. McKay, T. Alexander, L. Bello, M. J. Biercuk, L. Bishop, J. Chen, J. M. Chow, A. D. Córcoles, D. Egger, S. Filipp *et al.*, Qiskit backend specifications for OpenQASM and OpenPulse experiments, arXiv:1809.03452.

[92] A. Dutt and V. Rokhlin, Fast fourier transforms for nonequispaced data, SIAM J. Sci. Comput. **14**, 1368 (1993).

[93] G. L. Bretthorst, *Bayesian spectrum analysis and parameter estimation* (Springer Science & Business Media, New York, 2013), Vol. 48.

[94] H. Lange, S. L. Brunton, and J. N. Kutz, From Fourier to Koopman: Spectral methods for long-term time series prediction, J. Mach. Learn. Res. **22**, 41 (2021).

[95] D. E. Shea, R. Giridharagopal, D. S. Ginger, S. L. Brunton, and J. N. Kutz, Extraction of instantaneous frequencies and amplitudes in nonstationary time-series data, IEEE Access **9**, 83453 (2021).