Quantum-probabilistic Hamiltonian learning for generative modeling and anomaly detection

Jack Y. Araz [*] and Michael Spannowsky [†]
Institute for Particle Physics Phenomenology, Durham University, South Road, Durham DH1 3LE, United Kingdom

(Received 8 July 2023; accepted 28 November 2023; published 21 December 2023)

The Hamiltonian of an isolated quantum-mechanical system determines its dynamics and physical behavior. This study investigates the possibility of learning and utilizing a system's Hamiltonian and its variational thermal state estimation for data analysis techniques. For this purpose, we employ the method of quantum Hamiltonian-based models for the generative modeling of simulated Large Hadron Collider data and demonstrate the representability of such data as a mixed state. In a further step, we use the learned Hamiltonian for anomaly detection, showing that different sample types can form distinct dynamical behaviors once treated as a quantum many-body system. We exploit these characteristics to quantify the difference between sample types. Our findings show that the methodologies designed for field theory computations can be utilized in machine learning applications to employ theoretical approaches in data analysis techniques.

DOI: 10.1103/PhysRevA.108.062422

I. INTRODUCTION

The Hamiltonian plays a crucial role in our theoretical understanding of a physical system. The dynamics of an isolated quantum system is governed by an effective Hamiltonian which indicates the interaction of the system's constituents. In many, arguably simple, cases, it is possible to determine the effective Hamiltonian through a set of theoretical considerations such as observing its interactions and using the underlying symmetries of the system. Often, however, it is challenging to derive the algebraic form of a Hamiltonian from theoretical considerations only. Hence, several Hamiltonian learning methods have been proposed by employing thermal or eigenstates [1–6], short-time evolutions [7–10], and data-driven approaches [11]. With recent technological developments, it has become possible to simulate the effective Hamiltonian that governs a quantum many-body system in an actual quantum device. Widely used methods such as the variational quantum eigensolver [12,13] and its generalization, the variational quantum thermalizer [14], have become the most promising algorithms for noisy intermediate-scale quantum devices [15].

There is a substantial yet often underappreciated similarity between the computational methods used for data analysis, e.g., in quantum machine learning and the theoretical description of quantum many-body systems. In both scenarios, one optimizes the variational parameters of a given *Ansatz* over an objective function. For the former, this is naturally the expectation value of a Hamiltonian, and for the latter, it is a loss function chosen for the nature of the problem. In machine learning (ML) applications, one usually chooses one- or multiqubit measurement with a Pauli operator and optimizes the probability of the expectation value of this operator for a set of qubits. However, this operator is not necessarily optimal for the optimization process, where it is only a subset of possible combinations of different operators. In this study we investigate the possibility of learning an optimal effective operator for the optimization process and the implications of this operator for the application.

In generative modeling, the aim is to learn the joint probability distribution between the target and the observed data, which enables the model to generate new data resembling the observed data. This requires representing the probability distribution of the data within a quantum device. The mixed states are an ideal surrogate for such representation since they form probabilistic mixtures of pure states. Additionally, mixed states attain the properties of both quantum and classical correlations, enhancing the representability of a given probability distribution. The likeness of a given probability distribution can be captured within a parametrized quantum circuit (PQC) as a thermal state of a modular Hamiltonian. Quantum Hamiltonian-based models (QHBMs) [14] have been proposed as generative models which split the learning process into two distinct parts. The first part is responsible for learning a modular Hamiltonian with the aid of a classical neural network for capturing the classical correlations within the data. The second portion consists of a PQC that constructs the learned Hamiltonian's thermal state. The aim is to approximate the probability distribution of the data by optimizing the learned thermal state with respect to the mixed state based on the data.

Motivated by the close methodical relation between data analysis and the simulation of field theories, i.e., the use of optimization methods on parametrized circuits, we propose to learn a Hamiltonian from data, according to the

[*] jack.araz@durham.ac.uk
[†] michael.spannowsky@durham.ac.uk

QHBM approach. Thus, we will demonstrate an end-to-end hybrid quantum-probabilistic optimization procedure to simultaneously learn the probability distribution and modular Hamiltonian for the data. We then use the learned probability distribution to generate new data and, as a case study, apply it to top-quark production at the Large Hadron Collider (LHC). Furthermore, we will use the learned Hamiltonian for anomaly detection, as we will show that both the expectation value of the learned Hamiltonian and the Hamiltonian-based time-evolution sequence discriminate between signal and background data samples.

Our findings in this study show that the optimization methods developed to simulate quantum many-body systems are easily transferable to data-analysis applications and can be used to integrate the theoretical foundations of quantum mechanics into ML techniques and vice versa.[1] Previous studies on Hamiltonian learning techniques are mainly based on learning the structure of a quantum state, e.g., with simulated data from a known Hamiltonian with certain noise [11]. The initial proposal of the QHBM [14] was designed to provide a hybrid learning algorithm. However, due to the complexity of the quantum systems, it is challenging for a classical computer to generate efficient enough samples. To alleviate that problem, various solutions have been proposed, such as approximated free-energy techniques [23] and replacing classical sampling of the quantum data with a quantum circuit that represents the sample itself [24]. Simulating thermal states are especially challenging due to their circuit depth requirements, which can be remedied via noise-assisted thermal state preparation [25]. Additionally, various density-matrix simulation algorithms have been proposed for error mitigation [26] as well as algorithms that are enhanced with classical postprocessing techniques to simulate nontrivial Hamiltonian [27]. Despite the plethora of applications for quantum simulation, such techniques have not been employed in data analysis methodology, where the closest application was building a covariance matrix for principal component analysis through density matrices [28]. Our implementation goes beyond quantum simulations and discusses the usage of such techniques for data analysis by representing generic data as a mixed quantum state of a learned operator and, in doing so, generating an abstract representation of the entire data set as a Hamiltonian.

This study is structured as follows. In Sec. II we outline the methodology that is adapted. Section III introduces the data set and preprocessing scheme, with a generative modeling exercise in Sec. III A and anomaly detection in Sec. III B. We present a summary and offer conclusions in Sec. IV.

II. QUANTUM HAMILTONIAN-BASED MODELS

In this section we first delve into the intricacies of constructing the quantum variational *Ansatz*, emphasizing the incorporation of data embedding (Sec. II A). We then elucidate the construction of the Hamiltonian, providing a rationale for the specific approach chosen (Sec. II B). Next we delineate the formulation of the objective function, drawing parallels to classical generative modeling for a clearer understanding (Sec. II C). We provide a synthesis of the model in Sec. II D.

A. Quantum variational *Ansatz* and data representation

The objective of generative modeling is to encapsulate the entire feature space within a single probability distribution function, enabling efficient sampling to replicate the underlying distribution. Quantum mechanics provides a natural framework for this representation through the concept of mixed states. In quantum theory, a mixed state is a composite entity formed by combining pure states or other mixed states, serving as a faithful representation of the probability distribution encompassing its constituent states. A mixed state can be represented as

$$\sigma = \sum_i p_i |s_i\rangle\langle s_i|, \tag{1}$$

where p_i is the probability of observing the state $|s_i\rangle$ in the mixed state σ. Hence $\sum_i p_i = 1$. If we aim to acquire a density-matrix representation of the feature space for data regeneration, a challenge arises due to the inherent nature of quantum circuits as pure state simulators. Embedding a mixed state directly in a quantum circuit is not straightforward. To circumvent this challenge, we can reinterpret the data samples as a probability distribution. Each feature within a data point possesses an associated occurrence probability, which can be effectively encoded onto the quantum circuit using binary values (0's and 1's) derived from sampling a Bernoulli distribution. By generating a sufficient number of such samples, the mean of the sample set corresponds to the occurrence probability of the feature. Through the learning of these samples, it becomes possible to reconstruct the correlation relationships between the features within the quantum circuit.

By this method a single data point will be represented as a collection of pure states which are sampled from a Bernoulli distribution B

$$|p_n\rangle_d^i \equiv |p_1, \ldots, p_n\rangle_d^i = B(p_1, \ldots, p_n), \tag{2}$$

where the state for the dth data point with n individual feature probabilities, given as p_i, has been represented as the sample i, drawn from the Bernoulli distribution. This state can be embedded in a quantum circuit by applying Pauli-X gates where the Bernoulli distribution results in 1. The combination of the entire data set in terms of sampled states can be represented as

$$|\mathcal{D}\rangle = \sum_{d\in\mathcal{D}} \frac{\alpha_d}{N} \sum_i^N |p_n\rangle_d^i, \tag{3}$$

where α_d is the weight of each data point within the data set $\mathcal{D}$. Here N represents the number of samples drawn from the Bernoulli distribution for each data point. Finally, the mixed state of the entire data set can be represented as $\hat{\sigma}_\mathcal{D} = |\mathcal{D}\rangle\langle\mathcal{D}|$ with appropriate normalization. Note that now a data point does not correspond to a single circuit measurement but a

[1] In previous studies, the generative modeling was used in the context of quantum generative adversarial networks [16–19], and anomaly detection was presented via a PQC [20] and quantum variational autoencoders [21]. See Ref. [22] for an example of quantum machine learning for particle physics.

stack of circuits with different binary inputs from a single Bernoulli distribution. Furthermore, one can learn the correlation structure by means of a variational *Ansatz* $\hat{U}(\phi)$, where ϕ represents the trainable parameters of the *Ansatz*.

B. Building the Hamiltonian

To facilitate the optimization process, it is imperative to establish a well-defined measurement protocol. This protocol must employ an operator or Hamiltonian capable of accurately encapsulating the entropic probability distribution associated with the targeted mixed state for acquisition. While one option involves the application of a preexisting Hamiltonian, such as the Ising model, it is essential to acknowledge that this approach inherently assumes specific correlation structures among the features. Alternatively, a more ambitious endeavor entails simultaneously acquiring both the complete Hamiltonian representation and the density matrix characterizing the underlying data through an optimization process.

Given the resource-intensive nature of learning a Hamiltonian, we have opted for the utilization of a classical neural network, which offers a more flexible structural framework. However, it is worth noting that while any neural network *Ansatz* can be employed for this purpose, the estimation of the partition function may pose computational challenges that exceed the allocated resources. The necessity of a partition function will become evident in the subsequent section.

Energy-based models (EBMs) [29] present an ideal choice for our task, as they inherently encompass the partition function. An EBM establishes a mapping between a state configuration and a scalar energy measure, defined as $E_\theta(v) := v \in \mathcal{V} \to \mathbb{R}$, where $v \in \mathcal{V}$ represents a spin configuration within the set of all possible configurations. Energy-based models are designed to determine the optimal energy by minimizing the marginal probability distribution of the states in $\mathcal{V}$, given by

$$p(v) = \frac{1}{\mathcal{Z}_\theta} e^{-E_\theta(v)}, \quad \mathcal{Z}_\theta = \sum_{v \in \mathcal{V}} e^{-E_\theta(v)}.$$

Computation of the energy and partition function for all possible state configurations is generally a formidable task. To address this, we employ Monte Carlo (MC) algorithms to sample states, accepting the most probable ones based on the partition-free acceptance rate defined as $\min\left(p(v_{n+1})/p(v_n), 1\right)$ for a randomly initialized state v_{n+1} and a previously chosen random state v_n. By doing so, the MC algorithm generates an ensemble of $|v_n\rangle$, i.e., Gibbs, states. This algorithm can be visualized by the pseudocode in Algorithm 1.

With a sufficient number of MC samples N^{MC}, we define a modular Hamiltonian as

$$\hat{\mathcal{K}}_\theta = \sum_{n=1}^{N^{\mathrm{MC}}} E_\theta(v_n)|v_n\rangle\langle v_n|, \tag{4}$$

where $|v_n\rangle$ represents normalized spin states determined by the MC algorithm and $E_\theta(v_n)$ corresponds to their energy as measured by the chosen EBM *Ansatz*. It is important to note that $\mathcal{K}_\theta$ is defined as a Hermitian operator. Using the modular Hamiltonian definition in Eq. (4), the expectation value of the

Algorithm 1: The MC algorithm to form the modular Hamiltonian.

```
initial state: v_0;
while n < N^MC do
    propose a state: v_n;
    if min(p(v_n)/p(v_{n-1}), 1) < Random number
     then
        add to the Gibbs state;
        increment n;
    else
        continue;
    end
end
```

dth data point can be defined as

$$\langle \hat{\mathcal{K}} \rangle_{\theta,\phi} = \frac{1}{N} \sum_{i=1}^{N} \langle p_n|_d^i \hat{U}(\phi) \hat{\mathcal{K}}_\theta \hat{U}^\dagger(\phi) |p_n\rangle_d^i, \tag{5}$$

where the mean expectation value has been computed by taking the mean of N samples taken from the Bernoulli distribution.

C. Objective function

The primary objective of this endeavor is to establish a dependable representation of $\hat{\sigma}_\mathcal{D}$ through the acquisition of a learned density matrix, denoted by $\hat{\rho}_{\theta,\phi}$. In classical probabilistic learning, the optimization process revolves around minimizing the Kullback-Leibler divergence D_{KL} to diminish the disparity between two probability distributions [refer to Eq. (B4)] [30]. In this context, the goal is to approximate $\hat{\rho}_{\theta,\phi} \simeq \hat{\sigma}_\mathcal{D}$. Extending this principle to our scenario, where we work with a given Hamiltonian and temperature, the Gibbs-Delbrück-Moliére variational principle [31] asserts that the most suitable objective function for this process is the free energy, defined as

$$\mathcal{F} = E - \frac{1}{\beta} \mathcal{S}(\hat{\sigma}_\mathcal{D}), \tag{6}$$

which is bounded by the actual free energy of the system. Here $\mathcal{S}(\hat{\sigma}_\mathcal{D})$ denotes the entropy of the data [refer to Eq. (B3)], β represents the inverse temperature, and E signifies the expectation value of the given Hamiltonian as defined in Eq. (5). Notably, both the Hamiltonian and the entropy are unknown prior to data analysis, posing a significant challenge. Recognizing that the free energy can also be expressed as the log-partition function, we can reformulate the entire expression as

$$\beta E + k_\beta \ln \mathcal{Z}_\theta \geqslant \mathcal{S}(\hat{\sigma}_\mathcal{D}), \tag{7}$$

where k_β is a constant related to the true entropy of the data and the Boltzmann constant. This inequality offers a more suitable objective function for our purpose. Since temperature and the Boltzmann constant lack physical significance in the context of data analysis, they can be employed as regularizers of the objective function, and for this study we consider them to be equal to one.

After looking at the left-hand side of Eq. (7), it becomes evident that the entropy of the data essentially manifests as the

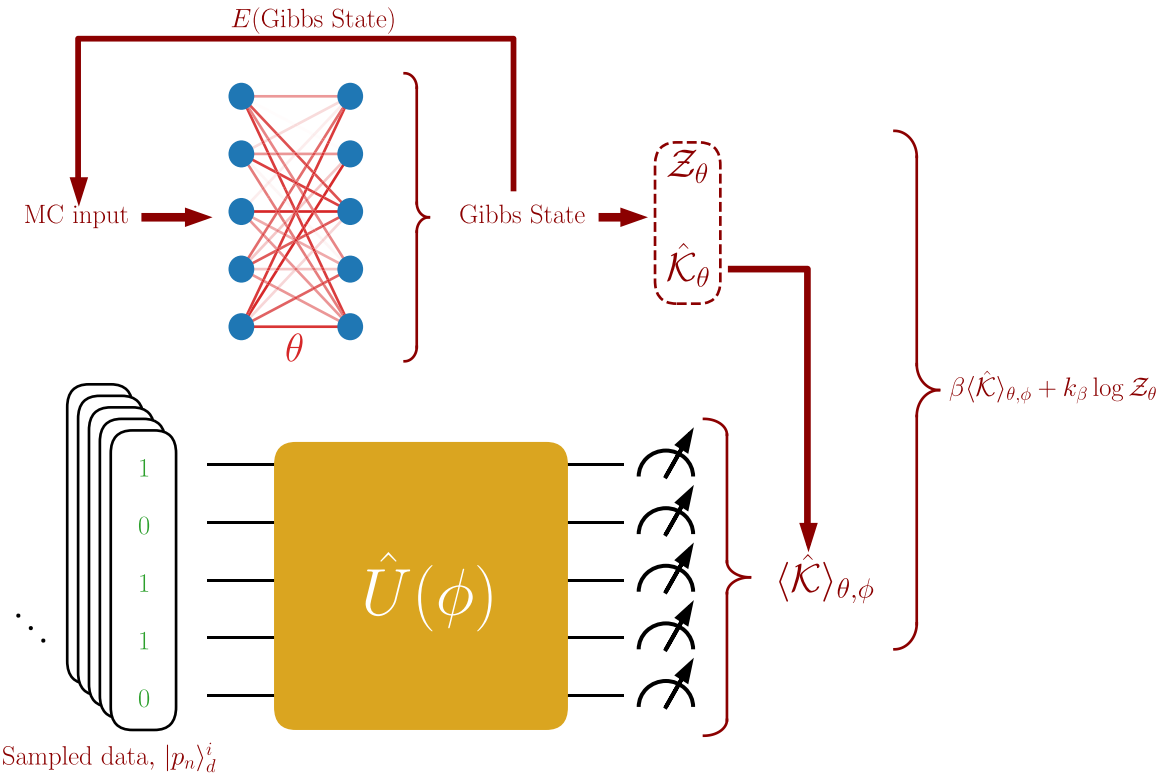

FIG. 1. Schematic representation of the quantum modular Hamiltonian-based learning for data analysis. Two parts of the implementation are represented as two parallel layers stacked on top of each other, with the top layer being responsible for forming a Hamiltonian by generating a Gibbs state through a MC algorithm based on an EBM. The bottom layer is responsible for computing the expectation value of the Hamiltonian for a sampled set of initial states. Finally, the expectation value and the partition function are combined to form the cost function of the network.

negative log-probability distribution of a multivariate Gaussian distribution centered at zero,

$$\mathcal{L}(\theta,\phi) = \frac{1}{\mathcal{Z}_\theta^{k_\beta}} e^{-\beta\langle\hat{\mathcal{K}}\rangle_{\theta,\phi}} \equiv \mathcal{N}(|\hat{U}(\phi)p_n\rangle|0,\Sigma(\theta)),$$

where the Hamiltonian can be interpreted as the covariance matrix $\Sigma(\theta)$ among different features. Given that the determinant of a Hermitian matrix is equivalent to the sum of its eigenvalues, $\mathcal{Z}_\theta^{k_\beta} \equiv \det[\Sigma(\theta)]^{\dim(\Sigma)}\sqrt{2\pi}$. Thus, our approach essentially models the data as a multivariate Gaussian distribution while simultaneously learning the covariance matrix through the optimization of $-\ln\mathcal{L}(\theta,\phi)$. Similar analogies can be found in the context of simulating lattice field theories using flow-based algorithms [32,33].

In unsupervised learning, the neural network serves as a statistical model of the underlying data and the objective is to minimize the negative log-probability distribution. Drawing from the analogy presented earlier, it becomes evident that reformulating this problem as a thermal state effectively implies an assumption that the data can be suitably approximated by a Gaussian distribution. This insight establishes a direct connection between theoretical approaches and conventional machine learning techniques, highlighting the interplay between sophisticated modeling strategies and established methodologies in the field of statistical thermodynamics.

D. Combining it all together

Figure 1 provides a schematic overview of the entire process, segmented into two primary sections. Illustrated on top is the generation of the modular Hamiltonian $\hat{\mathcal{K}}_\theta$ and the partition function $\mathcal{Z}_\theta$ through the utilization of a MC algorithm. In contrast, the bottom depicts the variational circuit, incorporating sampled data points as specified in Eq. (2) as input. Leveraging the modular Hamiltonian, we compute the expectation value for each data point, as detailed in Eq. (5). Subsequently, we combine the partition function and the expectation value to formulate the loss function, as elucidated in Eq. (7).

Once the mean loss function is computed for a batch of data points, we update the trainable parameters θ and ϕ using the expressions

$$\theta' = \theta + \eta\frac{\partial}{\partial\theta}\ln\mathcal{L}(\theta,\phi), \quad \phi' = \phi - \eta\beta\frac{\partial\langle\hat{\mathcal{K}}\rangle_{\theta,\phi}}{\partial\phi},$$

where η denotes the learning rate. Algorithm 2 shows a single training step for a batch of data. Within the concept of batched learning, optimizing a negative log-probability distribution has two significant computational bottlenecks. First, as mentioned above, one must estimate a proper modular Hamiltonian for the optimization procedure. This has been done independently of the input data where the MC algorithm determines the most probable set of spin configurations and, by doing so, automatically minimizes the energy of the EBM

Ansatz. This requires reestimation of the modular Hamiltonian after each update. It is important to emphasize that since the MC algorithm proposes an input configuration completely independent of the input data, the optimization procedure is the only connection between data and the Hamiltonian. The second bottleneck involves $\hat{\sigma}_d$ estimation; since we do not have access to quantum data, we need to sample through the probability distribution of each data point and compute the expectation value $\langle\hat{\mathcal{K}}\rangle_{\theta,\phi}$.

Algorithm 2: Pseudocode for a single training step.

```
Form K̂_θ and Z_θ with algo. 1;
Number of Bernoulli samples: N_b;
Probability of all features in a data point: p;
while i < number of samples in a batch do
    while n < N_b do
        generate n-th sample from i-th data point;
        compute ⟨p|_n^i K̂_θ |p⟩_n^i of n-th sample;
        increment n;
    end
    take the mean of N_b expectation values;
    increment i;
end
take the mean of the expectation values in batch;
compute loss with mean batch expectation value
  and partition function;
compute gradient of the loss;
update θ and φ;
```

Such an optimization process allows the quantum circuit to learn a nonlinear distribution by minimizing $E_\theta(v_n)\langle\hat{\sigma}_\mathcal{D}\hat{U}(\phi)||v_n\rangle\langle v_n||\hat{\sigma}_\mathcal{D}\hat{U}(\phi)\rangle$. Since $|v_n\rangle$ is constructed from a nonlinear classical neural network, $|\hat{\sigma}_\mathcal{D}\hat{U}(\phi)\rangle$ is being forced to approximate such nonlinear behavior to reduce the distance of the projection.

Although it is a powerful representation of the data, learning a completely free Hamiltonian is computationally challenging simply because the Hamiltonian has to be decomposed into Pauli operators at every step of the optimization process. It is possible to avoid the EBM if we assume a certain structure for the modular Hamiltonian $\hat{\mathcal{K}}$. For instance, a generic Hamiltonian that captures the nearest-neighbor interactions can be suitable to capture near-term complexity of the data,

$$\hat{\mathcal{K}}_\theta = \sum_{i\in\text{qubits}} \theta_{i,i+1}(\sigma_i^+\sigma_{i+1}^- + \sigma_{i+1}^+\sigma_i^-),$$

where σ^+ and σ^- are raising and lowering operators, respectively, and θ is the trainable coupling strength. Since the summation captures only nearest-neighbor interactions, this Hamiltonian may not be able to capture the complexity of the data, but it will simplify the optimization process significantly. Additionally, tensor network techniques can aid in the decomposition of large Hamiltonian matrices. However, such simplifications are beyond the scope of this paper, where we focus on the most generic application and see if it is indeed possible to create a useful operator through this procedure.

III. RESULTS

As a case study, we use a top tagging data set [34,35], which includes over 10^6 mixed collider events for semileptonic top and dijet production channels at $\sqrt{s} = 14$ TeV. Events are generated and showered in PYTHIA 8 [36], and the detector simulation is achieved using DELPHES 3 package [37] with a default ATLAS configuration card. All jets are reconstructed via the anti-k_t algorithm [38] with $R = 0.8$ within the FASTJET [39] package. Furthermore, the central-boosted phase space is captured by requiring the jet transverse momentum p_T to be within [550, 650] GeV and absolute pseudorapidity to be $|\eta| < 2$.

The jets are further processed to be represented as calorimeter images, potentially captured by the hadronic calorimeter in the LHC experiment. Following the procedure presented in Refs. [40,41], leading jet constituents are centered around the jet axis on the pseudorapidity–azimuthal angle η-ϕ plane within $[-1.5, 1.5]$. Each image is divided into four quadrants and the most energetic quadrant has been moved to the top right corner by horizontally and vertically flipping the image. Finally, all the training samples are standardized over randomly chosen 200 000 images by fitting p_T within the $[0, \pi]$ range. This standardization procedure yields calorimeter images of 37×37 pixels; however, since it is not possible to process this within a quantum circuit, we simplify our data by cropping 12 pixels from each axis and downsampling the resulting image by taking the mean of four adjacent pixels.

Figures 2(a) and 2(b) show the mean of 5000 images for the signal and background, respectively, where η' and ϕ' are the modified pseudorapidity and azimuthal angle axes after standardization. The color represents the value of the transverse momentum in each pixel, measured in Fig. 2(d). Note that this is before normalizing the p_T distribution within the $[0, \pi]$ range. The following two images capture only a single event within the signal [Fig. 2(c)] and background [Fig. 2(d)] samples. All the images are cropped to show only a 27×27 central image to focus on the main activity. Even though the averaged images are easily differentiable, single events are usually random looking and not easily differentiable; hence various ML techniques have been developed to differentiate these samples.

Because the top decays into two light jets through a W decay and a b jet, it creates a three-prong signature, as shown in Fig. 2(a). Such topological behavior has been exploited by many analytic tagging algorithms (see, e.g., Ref. [42]). The dijet signature, on the other hand, leaves a single-prong signature on the calorimeter, as shown in Fig. 2(b). This is due to the fact that dijet events do not contain enough energy to produce two distinct jet signatures. Such a process is crucial to investigate at the LHC because the top quark's mass can further our understanding of the Higgs mechanism and its coupling to the top quark since mass comes with a large coupling to the Higgs boson. Even with larger center-of-mass energies, the production of top quark pairs has been improved at the LHC. These events are usually contaminated with dijet events, making it challenging to isolate top quark events. Hence it is vital to separate top events from the dijet background to improve the experiment's sensitivity to its couplings.

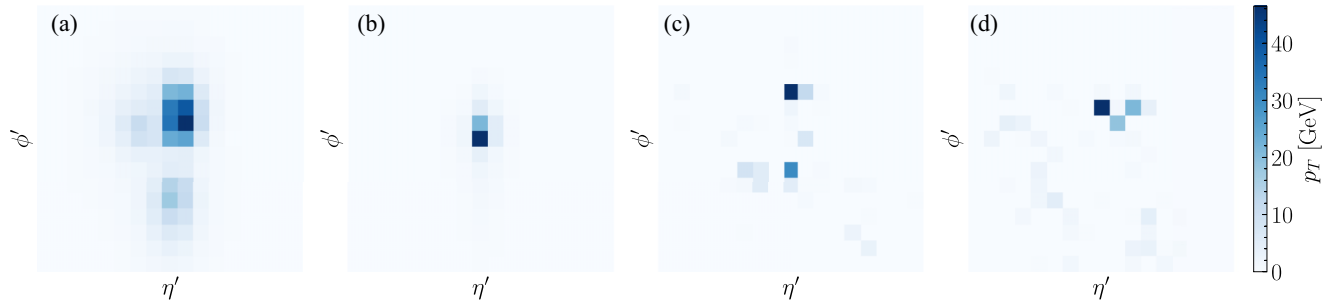

FIG. 2. Signal and background images projected on the η'-ϕ' frame: (a) signal and (b) background represented with a mean of 5000 randomly chosen events and (c) signal and (d) background with a single event for each sample. For this representation, ten pixels have been cropped from each axis of the images from the original 37×37 pixels. Color represents the magnitude of energy deposited in each pixel.

In the following sections we will use only a fraction of these images by cropping and downsampling them due to computational limitations. This mainly affects the span of information since it is highly dependent on the geometrical positions of the energy deposits on each pixel. We will start with the central pixels and increase the pixel count from there, but it is important to note that the central four pixels are maximally similar throughout both samples. Hence one includes more information regarding the nature of the event once we go beyond the central four pixels.

A. Generative modeling

As the first set of applications for the QHBM, we will aim to learn the probability distribution of the pixel intensity in calorimeter images. Each standardized sample pixel has p_T intensity within $[0, \pi]$. The pixel intensity can be interpreted as probability distribution if it passes through a bijective function, which outputs values in the range [0,1], such as a sigmoid function. This will allow us to simultaneously interpret pixel intensities as probability distributions and convert them back to their status quo. Due to the computational cost of the quantum simulation and the optimization methodology, we choose to perform our investigation with only the central four pixels, which retain the necessary information to differentiate between top and dijet images as presented in a previous study [43].

The modular Hamiltonian is determined via a restricted Boltzmann machine (RBM), with details presented in Appendix A. We reestimate the modular Hamiltonian for each batch training by collecting a set of spin states via the MC algorithm presented above. The initial state for each training is set to $|\uparrow \cdots \uparrow\rangle$; each following MC algorithm is initiated by the last state determined in the previous MC run. For each execution the MC algorithm runs for 100 steps to converge on a stable Gibbs state without collecting any; the number of collected states is analyzed case by case below. Note that these states are entirely independent of the input data; hence the MC algorithm independently minimizes the energy of the RBM by determining the most probable set of states.

The expectation value for each image is estimated via Eq. (5). Since we are employing batched learning, the expectation value of the batch is computed by taking the mean of each expectation estimation in the batch. Finally, the variational parameters of the network are updated with respect to the mean objective function

$$\arg\min_{\theta,\phi} \frac{1}{N_{\text{batch}}} \sum_{i}^{N_{\text{batch}}} -\ln \mathcal{L}(\theta, \phi|\hat{\sigma}_i).$$

Note that the mean only entitles the expectation value of the modular Hamiltonian. We divided our study into different benchmarks to study the effects of $\hat{\sigma}_{\mathcal{D}}$ and $\hat{\mathcal{K}}_\theta$ estimations. The PENNYLANE package [44] is employed for quantum circuit simulation; the RBM and optimization are held within TENSORFLOW [45,46] and TENSORFLOW-PROBABILITY [47] packages. Our implementation can be found in [48]. All the benchmarks are trained with 1000 training samples, and overtraining is monitored with the same number of validation events.[2] The Adam optimization algorithm [49] is employed with a 10^{-2} initial learning rate, with the learning rate reduced by half if validation loss does not improve for over 25 epochs. Each benchmark is trained for 100 epochs, and training is terminated if the validation loss does not improved for over 50 epochs.

For the quantum *Ansatz*, we use a matrix product state (MPS) structure [50] where two-qubit operators are applied to each adjacent qubit in a staircaselike architecture, which is depicted in Fig. 3. We refer to each of these constructions from the first qubit to the last as a layer. Each two-qubit operator $\hat{U}_i(\phi)$ includes two rotation gates around the Pauli-Y axis for each input qubit with an independent variational rotation angle followed by a controlled-NOT gate. For each benchmark, we use three layers. Note that the algorithm is also tested with different architectures such as a simplified two-design [51] and strongly entangling layers [52], which is observed to improve the results.

Figure 4 shows the test metrics for each benchmark where each point is tested with 10 000 mixed test events and presented with one standard deviation, estimated by dividing the test sample into batches of 25. The left column shows the benchmarks for $\hat{\sigma}_{\mathcal{D}}$ estimation where the N^{MC} for $\hat{\mathcal{K}}_\theta$ estimation is set to 200, while the right column shows the benchmarks for $\hat{\mathcal{K}}_\theta$ estimation where the N^{smp} for $\hat{\sigma}_{\mathcal{D}}$ estimation is set to 5000. It is also important to note that the

[2]It is essential to note that we did not observe any significant improvement in generalization for more extensive training sets; hence, due to the computational cost, we limited the analysis to 1000 events.

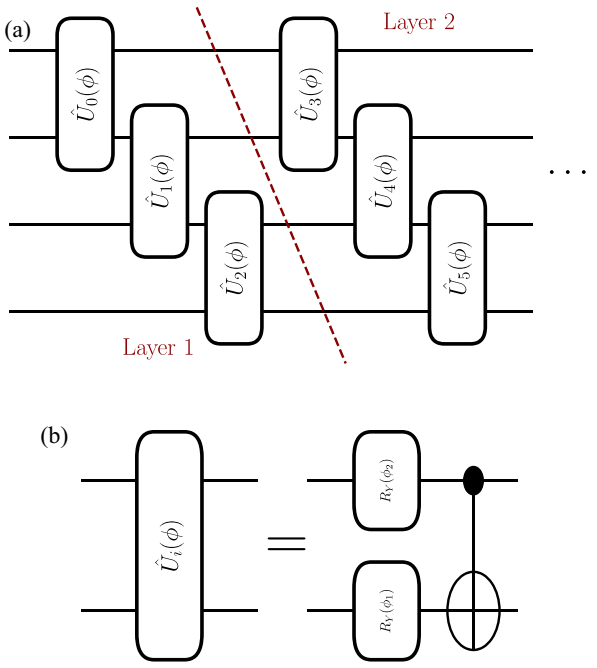

FIG. 3. (a) Schematic diagram of the MPS variational circuit *Ansatz* and (b) structure of each $\hat{U}(\phi)$ gate that has been used.

samples to estimate $\hat{\sigma}_{\mathcal{D}}$ for the right column are generated before the optimization process to speed up the application; however, for the left column, each sample produced during the training effectively allows those benchmarks to experience different samples in each iteration. Figures 4(a) and 4(b) show the Kullback-Leibler distance between input images (q) and the sampled output images (p) and Figs. 4(c) and 4(d) the batched mixed state of the data ($\hat{\sigma}_{\mathcal{D}}$) and estimated mixed state ($\hat{\rho}_{\phi}$).[3] Figures 4(e) and 4(f) present the trace distance and Figs. 4(g) and 4(h) the fidelity between the thermal state of the data and the estimated thermal state. Finally, Figs. 4(i) and 4(j) plot the estimation of the von Neumann entropy separately for the signal [blue (dark gray)] and the background [red (light gray)]. Note that for the thermal state of the data all images in the input are assigned the same weight, i.e., α_i in Eq. (3). For details about these metrics, we refer the reader to Appendix B.

During our tests, we observed that the performance of the generative model is mainly based on the wellness of the estimation of the mixed state of the data, where for larger samples we observed improved the fidelity and trace distance (see the left column of Fig. 4). Furthermore, we observed that the wellness of the estimation also improves the Kullback-Leibler distance between the input and estimated states and exponentially reduces this metric's uncertainty. Note that $\mathcal{S}(\hat{\sigma}_{\mathcal{D}})$ is presented separately for the signal and background. Although we do not see any significant difference in signal or background for any other metric, the entropy for different sample sets is clearly separated. Note that all benchmarks are trained with mixed data and they are not exposed to the information regarding the data type.

In the right column of Fig. 4 we present the effect of the $\hat{\mathcal{K}}_{\theta}$ estimation on the same test metrics. Although we do not observe any significant improvement in fidelity, trace distance, and Kullback-Leibler distance [except a minor refinement in $D_{\mathrm{KL}}(q|p)$], we observe that the wellness of $\hat{\mathcal{K}}_{\theta}$ estimation improves the entropy estimation of the data and reduces the uncertainty. Hence Fig. 4(j) indicates that for good enough $\hat{\mathcal{K}}_{\theta}$ and $\hat{\sigma}_{\mathcal{D}}$ estimation, signal and background samples will produce unique entropy values. Thus this information can also be used to identify the nature of the data. However, $\mathcal{S}(\hat{\sigma}_{\mathcal{D}})$ is not observed to be a powerful discriminator. We compute the receiver operating characteristic curve to quantify the difference between the signal and background, and the highest area under the curve value we observe is around 0.7.

Note that we have not discussed the advantage of learning an operator for the data. In the following section we will discuss a possible usage of the modular Hamiltonian in the context of anomaly detection.

B. Anomaly detection

Anomaly detection is a methodology in which the network *Ansatz* learns the structure of the known data and tries to detect the difference in new data, if any. For this purpose, we use two test cases. For the first case, we use six qubits, where in addition to the central four pixels we add the top two pixels into the collection. For the second case, we also include the bottom two pixels to test the algorithm for the eight-qubit scenario.

We are using the same procedure outlined in Sec. III A, training both scenarios using background-only samples for 100 epochs and 1000 events with $\hat{\sigma}_{\mathcal{D}}$ estimated by 5000 samples before the training. The only difference between the two test cases is that we use 500 MC samples for the six-qubit scenario and 1000 for the eight-qubit scenario. The difference is due to the size of the latent space, where we observe that a larger latent space requires more MC samples to estimate $\hat{\mathcal{K}}_{\theta}$ for the stability of the result, which we will discuss later in this section.

The network results are tested with 10 000 background-only test samples. For the six-qubit scenario, we observe a fidelity of 0.81 and a trace distance of 0.3, whereas for the eight-qubit scenario we observe 0.79 and 0.3, respectively.

Although von Neumann entropy, as shown in Sec. III A, can lead to a significant observable to differentiate two types of samples, we propose a different observable based on the modular Hamiltonian. We will analyze two different cases; for the first, we will look into the effect of time evolution. We will define the time-evolution operator of a modular Hamiltonian as

$$e^{-iT\hat{\mathcal{K}}_{\theta}} \simeq \prod^{N} e^{-i\Delta t\hat{\mathcal{K}}_{\theta}} \equiv \mathcal{T}_N,$$

where $T = N\Delta t$. For small Δt, this operator can be applied in the quantum circuit under the Trotter-Suzuki approximation. Using this relation, one can compute the fidelity $\mathbb{F}$ of the

[3]Note the difference in notation here. For the variational thermal state we use $\hat{\rho}_{\theta,\phi}$ during the training since the density matrix at this stage is influenced by both the EBM and $\hat{U}(\phi)$. However, during the testing, the modular Hamiltonian is not used; thus, the variational thermal state is only influenced by $\hat{U}(\phi)$.

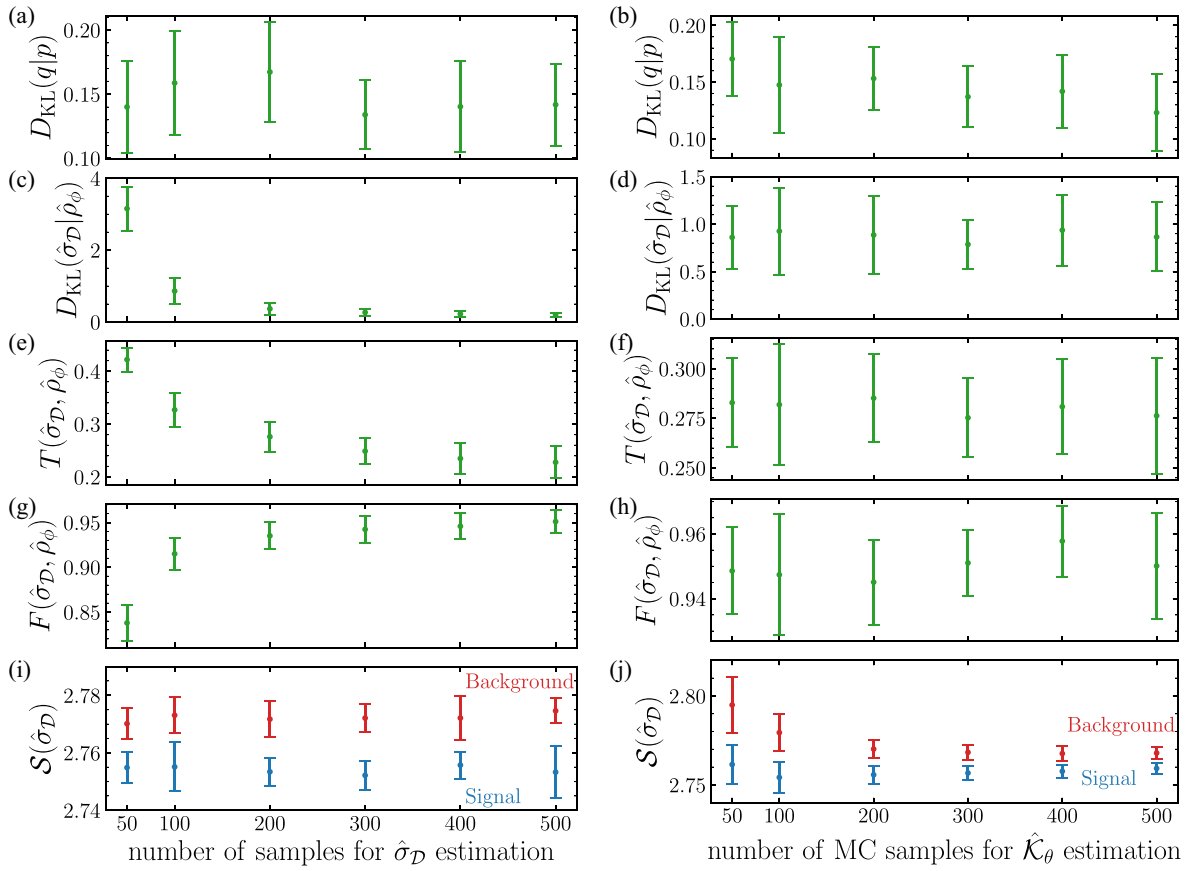

FIG. 4. Test metrics for (a), (c), (e), (g), and (i) the network trained with a different number of samples for density-matrix estimation and (b), (d), (f), (h), and (j) the network trained with a different number of MC samples for $\hat{\mathcal{K}}_{\theta}$ estimation, showing (a) and (b) the KL divergence between input and output samples, (c) and (d) the KL divergence, (e) and (f) the trace distance, (g) and (h) the fidelity between the truth level density matrix and the network's density-matrix estimation, and (i) and (j) the network's estimation for von Neumann entropy, where red (light gray) and blue (dark gray) represent background and signal samples. The results are prepared using 200 MC samples for the estimation of the Hamiltonian (left column) and 5000 samples for $\hat{\sigma}_{\mathcal{D}}$ estimation (right column). Each result is presented with one standard deviation, estimated by dividing 10 000 test samples into batches of 25 events.

time-evolved quantum state as

$$\text{Fidelity} = \langle\psi(t)|\psi(0)\rangle, \tag{8}$$

where $|\psi(0)\rangle = \hat{U}(\phi)|p_n\rangle_d^i$ and $|\psi(t)\rangle = \mathcal{T}_N|\psi(0)\rangle$. For the second case, since it is computationally less costly, we will analyze the expectation value without time evolution.

We computed the time evolution up to $T \leqslant 500$ for estimated $\hat{\mathcal{K}}_{\theta}$ in each scenario with $\Delta t = 0.1$ time steps. Figures 5(a) and 5(c) show the fidelity (8) concerning each time step for signal (blue) and background (red) samples, for the six-qubit scenario [Fig. 5(a)] and the eight-qubit scenario [Fig. 5(c)]. The thickness of each curve shows one standard deviation for the entire test sample.[4] Note that for the sake of visibility, Fig. 5(a) is limited to $T \leqslant 200$, while Fig. 5(c) is for $T \leqslant 500$. In order to devise a quantitative measure, we computed the power-frequency curve from the fast Fourier transform of the time-evolution sequence [see Eq. (B5)]. For the mean time-evolution sequence, we present the power-frequency distribution in Figs. 5(b) and 5(d) for each respective time-evolution result. Although we do not observe any significant difference in low-frequency regions, the power of both curves becomes significantly different for high-frequency regions. It is essential to note that the power-frequency curves become identical once the network is trained with mixed signal and background samples. Additionally, for the four-qubit scenario, the differentiability is observed to be significantly low. We computed the receiver operating characteristic (ROC) curve concerning the power distribution for a frequency threshold to quantify the ability to differentiate between two samples via the time-evolution sequence. The true (false) positive rate, i.e., signal (background) efficiency, has been computed by counting the number of events in the binned power distribution between its maximum and minimum values for a given frequency. Figure 6(a) shows the ROC curve and corresponding area under the curve (AUC) values for six-qubit (blue) and eight-qubit (red) scenarios. The black dashed line shows the random choice where the classification quality improves as the curves move further away from this

[4]Note that the test sample is limited due to the high computational cost.

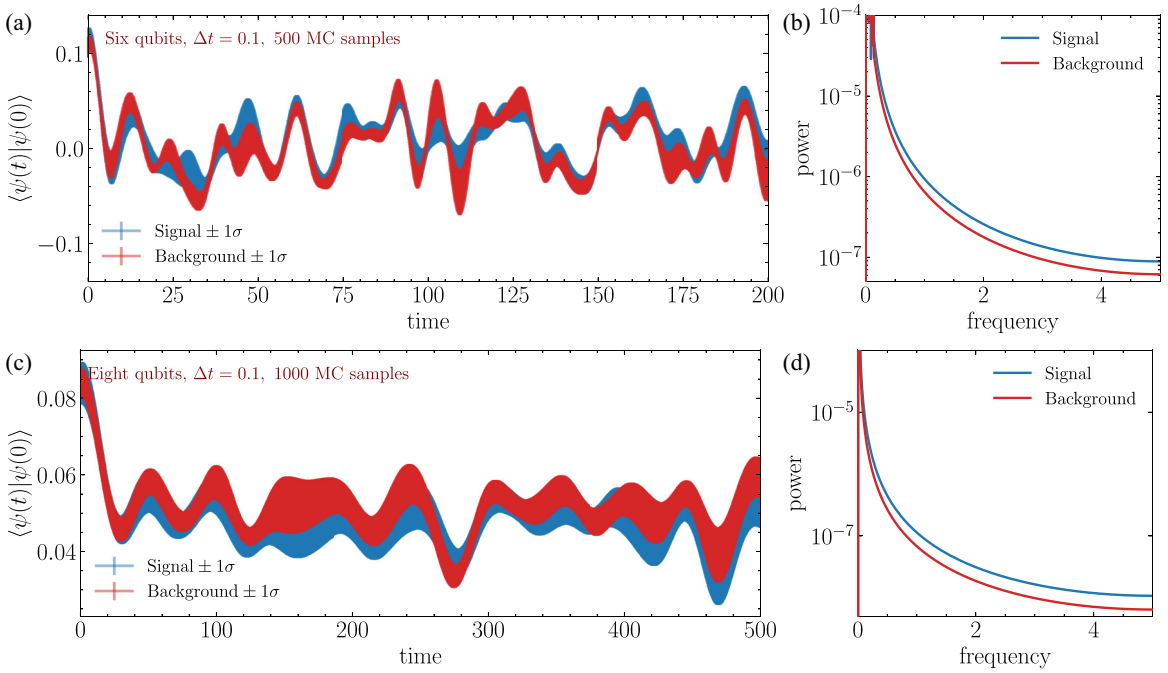

FIG. 5. Time evolution of the modular Hamiltonian for (a) and (b) six-qubit and (c) and (d) eight-qubit scenarios for (a) and (c) the fidelity distribution and (b) and (d) the power spectrum of the FFT of the distributions. The signal and the background are represented by blue (dark gray) and red (light gray), respectively.

line towards the top left corner of the plot. The best minimum frequency value has been chosen for both distributions; hence we did not observe any improvement in the AUC value for larger frequencies. We observe that the eight-qubit scenario reaches saturation at a frequency of 0.056 with a 0.85 AUC value. In contrast, the six-qubit scenario requires a frequency of 0.2 to reach saturation at a 0.82 AUC.

For the second, less costly, method we compared the expectation value for the signal and background without any time-evolution step $T = 0$. Figure 6(b) shows the ROC curve computed for 200 different thresholds chosen between maximum and minimum expectation values. We tested the results for a 10 000-event signal and background test sample where, as before, the red and blue curves show the results for eight- and six-qubit scenarios, respectively. Even at the initial time step, we observe that AUC values for both cases are above 0.9.

Utilizing the time-evolution sequence, we observe up to a 3% difference between the six- and eight-qubit scenarios, reducing the required frequency by 72%. Note that we are barely able to achieve 50% using a four-qubit scenario with the largest frequency that we compute; thus, adding new information significantly affects the ability to differentiate two sequences. Using only the information from the expectation

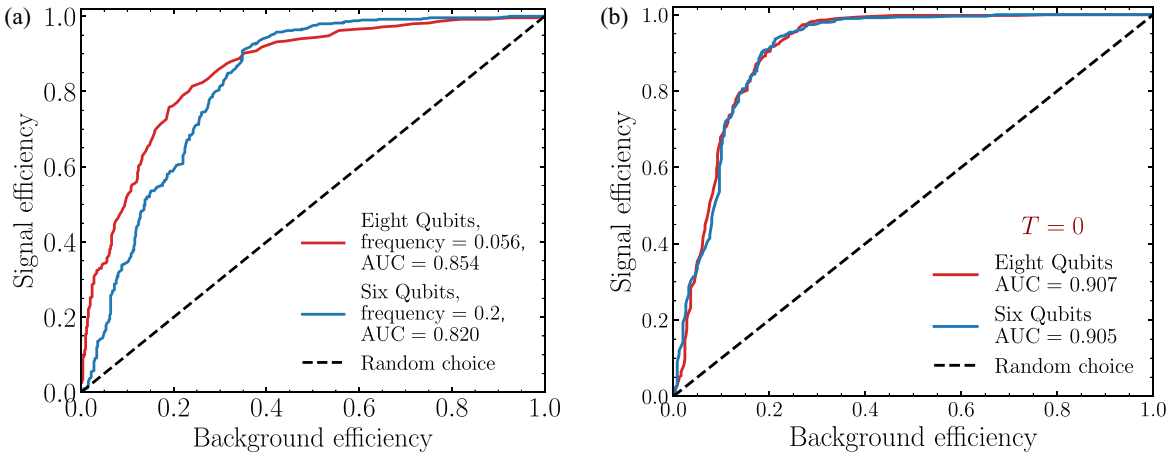

FIG. 6. ROC curve computed for the two types of measure used in this study. Results are shown for (a) the time-evolution sequence of the learned Hamiltonian and (b) the expectation value at $T = 0$. The red (light gray) and blue (dark gray) solid curves represent eight- and six-qubit scenarios and the black dashed curve shows the random choice.

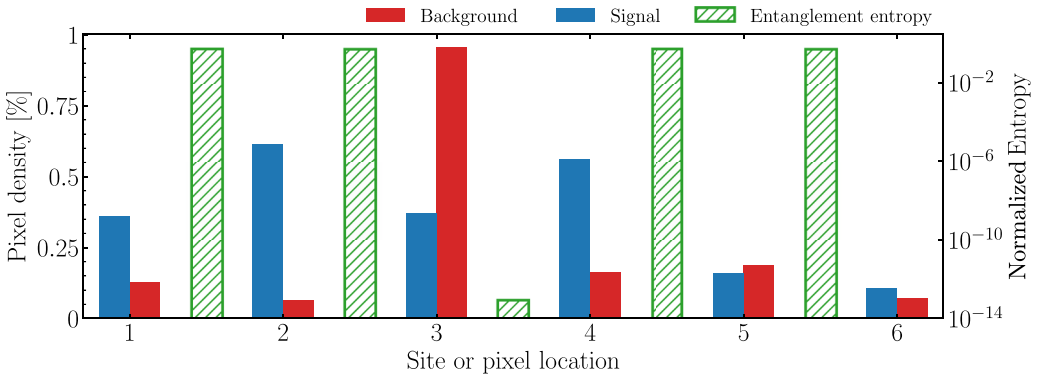

FIG. 7. Solid bars show the pixel density of each site for all the test data, with the value bound to the left y axis. Crosshatched bars show the relative entropy, computed from the lowest eigenvector of the learned Hamiltonian, between each site and the value which is bound to the right y axis. The x axis shows the site or pixel location.

value provides significantly better differentiability, whereas in the six-qubit (eight-qubit) scenario we observed 9% (6%) improvement in AUC values.

As mentioned before, the stability of the results relies on sufficient MC samples for $\hat{\mathcal{K}}_\theta$ estimation. Due to the probabilistic nature of the EBM, the computation of $\hat{\mathcal{K}}_\theta$ leads to a slightly different modular Hamiltonian. Hence the stability depends on increasing the number of samples; in other words, it depends on reaching a stable Gibbs state. For a lower number of MC samples, we observed a more significant standard deviation in each sample and lower differentiability between two sets of samples where the AUC value was significantly lower.

This exercise shows that the data from different sources can be interpreted as distinct quantum states; hence their corresponding Hamiltonian will produce different results when it acts on different states produced by these data samples. Since the Hamiltonian should be able to capture the entropic probability density of the given data, we investigate von Neumann entropy between each site at the ground state of the learned Hamiltonian. The reason for using von Neumann entropy is that it captures the information flow between reduced density matrices, and the change in the entropy value indicates statistically viable information for the optimization process. This measure has also been utilized in Ref. [41] to compress the feature space with an MPS *Ansatz*. The von Neumann entropy has been computed by first finding the lowest eigenvector of the six-qubit learned Hamiltonian via direct diagonalization. Furthermore, we constructed the reduced density matrix between two sites corresponding to each pixel. Figure 7 shows the pixel density averaged over the test set for signal (blue) and background (red) bars captured by the left y axis. This shows which pixels are statistically more active. The green crosshatched bar shows the relative entropy between two sites, where the right y axis has captured the value. The x axis shows the location of each pixel on the circuit and the green bars are placed in between each pixel location. We observe that the entropy values remain high between the low-density pixels. However, we observe exponentially low-entropy values between pixels 3 and 4, where pixel 3 has the highest density among all background pixels. It is essential to emphasize here that the learned Hamiltonian does not have any access to the input data and it is constructed by generating a Gibbs state through an MC algorithm. Hence the only link between the data and the Hamiltonian is the optimization algorithm, which enables the Hamiltonian to capture the statistical distribution of the input.

IV. DISCUSSION AND CONCLUSION

Quantum Hamiltonian-based models are a group of *Ansätze* that attempts to approximate the probability distribution of the data by representing it as a thermal state of a learned Hamiltonian. In this context, the computationally intensive Hamiltonian learning has been mitigated to a classical network, and a variational quantum circuit has been optimized with respect to the expectation value of the learned Hamiltonian. This method is a generalization over the variational quantum thermalizer technique, where one generates the thermal state of a given Hamiltonian at a target temperature. However, using a specific Hamiltonian for an ML application will be highly constrained since it is not always possible to *a priori* know the correlation structure of a given data. Hence, it has been learned during the optimization process by utilizing a classical energy-based model. This enables us to create a unique Hamiltonian for the data, which can then be used to scrutinize the properties of the data further. Thus this study demonstrates that the methods developed for quantum simulations are flexible and reusable for ML applications, hence showing the strong link between theoretical approaches and statistical ML techniques. This can lead to a more interpretable and intuitive *Ansatz* by virtue of our knowledge of quantum theory, and this study has aimed to take a step further to achieve a fully theory-driven ML technique.

In this study we demonstrated the usage of the QHBM for generative learning and anomaly detection for LHC data. We showed that the calorimeter images could be embedded in quantum circuits as a mixed state, and a variational thermal state of a learned Hamiltonian can represent their probability distribution. As a by-product of the optimization process, the objective function converges to the entropy of the data, which has been observed to produce unique values for different types

of data. Hence, this information can be further used to identify the generated data samples.

It is essential to ask if it is possible to use the learned Hamiltonian to understand the data structure further. We have presented two possible use cases of the learned Hamiltonian for anomaly detection. For the first case, we analyzed the expectation value of the time-evolution sequence for the learned Hamiltonian. We showed that by converting the sequence to the frequency domain, one could observe significantly different curves for two types of samples by computing the power distribution for the fast-Fourier-transformed sequence. Second, we showed that even the expectation value of the learned Hamiltonian is significantly different for different data types, which we quantified by analyzing the difference at various thresholds.

Our findings signify a fundamental property of the quantum many-body Hamiltonian. Once learned, the given Hamiltonian represents the dynamical properties of a specific quantum state. Since signal and background samples form significantly different state representations, a Hamiltonian designed for one type of sample reacts differently to a different system, since these systems have distinct dynamical properties. Hence we showed that it is possible to treat a given data sample as a quantum many-body system, and by using theory-driven optimization techniques, one can learn this system's Hamiltonian to be used to understand its properties. Although we only showed two possible use cases for generative modeling and anomaly detection, we hope that such approaches can be taken to devise more interpretable ML applications and build dedicated optimization algorithms that can utilize the system's physical properties.

Although the usage of the quantum theory comes with significant advantages, it is essential to admit that this method comes with undeniable computational costs and limitations. The obvious problem is the ability to execute these quantum circuits within a quantum device. Although we used a relatively small number of qubits, since generating the mixed state of each data point within the circuit requires many executions, we could not reproduce these results with a current quantum device. However, this can be improved by storing the input mixed states within a quantum memory device, which alleviates the need to regenerate such a computationally expensive process. As presented in the anomaly detection example, for this particular data set, the geometrical position of the active pixels is crucial to characterize the data set. Hence increasing the number of qubits will allow more information, and our experiments indicated that it would allow for the simulation of fewer time steps for discrimination. Increasing the number of qubits also requires the implementation of extensive correlations between features. An MPS *Ansatz* was suitable enough since our experiments were implemented with a few features. Still, we observed significant gains when more complex circuit architectures were implemented, which will be increasingly important with the implementation of larger feature spaces and makes the classical computation of the circuit increasingly challenging.

A further obstacle to the method is the completely free modular Hamiltonian which significantly affects the algorithm's scalability. With the increasing qubit size, the modular Hamiltonian grows exponentially via $2^{N_q} \times 2^{N_q}$, which makes it quite challenging to scale the algorithm for larger systems. As we discussed before, this can be avoided by imposing certain assumptions on the modular Hamiltonian to limit its shape.

ACKNOWLEDGMENTS

We thank Ongun Arısev and Soner Albayrak for discussions. J.Y.A. acknowledges the Galileo Galilei Institute.

APPENDIX A: RESTRICTED BOLTZMANN MACHINE

The restricted Boltzmann machine is a generative network which learns the joint probability distribution that maximizes the log-likelihood function [53–55]. Compared to the generic Boltzmann machines, the RBM is formed as an undirected, asymmetrical bipartite graph with two layers, i.e., visible and hidden, where all visible nodes are connected to all hidden nodes. The energy of the RBM is defined as

$$E(v,h) = -\sum_i \mathcal{B}_i^{\text{vis}} v_i - \sum_j \mathcal{B}_j^{\text{hid}} h_j - \sum_{i,j} v_i h_j \mathcal{W}_{ij}, \quad \text{(A1)}$$

where $\mathcal{B}^{\text{vis}}$ and $\mathcal{B}^{\text{hid}}$ stand for visible and hidden biases, respectively, $\mathcal{W}_{ij}$ is the weight matrix between visible and hidden state, and h and v stand for hidden (or latent) and visible (or input) states, respectively. Each configuration of the visible state is associated with a scalar energy measure [Eq. (A1)], which measures the compatibility of a given visible state where high-energy stands for low compatibility. The goal of an energy-based model is to minimize the predefined energy function.

A hidden state is constructed with respect to the given visible state where the probability of the hidden state being one is given as

$$p(h|v;\theta) = \sigma(v\mathcal{W} + \mathcal{B}^{\text{hid}}),$$

where θ stands for the collection of the trainable parameters presented in $\mathcal{W}$, $\mathcal{B}^{\text{hid}}$, and $\mathcal{B}^{\text{vis}}$ and σ stands for the sigmoid function. In order to construct hidden states h, one samples from a Bernoulli distribution with probability $p(h|v;\theta)$. Similarly, the reconstruction probability of the visible state is given by

$$p(v|h;\theta) = \sigma(h\mathcal{W}^T + \mathcal{B}^{\text{vis}}).$$

APPENDIX B: METRICS

The fidelity of two matrices is given by

$$F(\sigma,\rho) = [\text{Tr}(\sqrt{\sqrt{\sigma}\rho\sqrt{\sigma}})]^2. \quad \text{(B1)}$$

The trace distance between two matrices is defined as

$$T(\sigma,\rho) = \tfrac{1}{2}\text{Tr}[\sqrt{(\sigma-\rho)^\dagger(\sigma-\rho)}]. \quad \text{(B2)}$$

The von Neumann entropy of a matrix is given as

$$\mathcal{S}(\sigma) = \text{Tr}[\sigma \ln(\sigma)]. \quad \text{(B3)}$$

The Kullback-Leibler divergence between two probability distributions (or two density matrices) is defined as

$$D_{\text{KL}}(p|q) = p\ln(p) - p\ln(q). \quad \text{(B4)}$$

The power P of the fast Fourier transform is defined as [56]

$$P(\lambda) = \mathrm{Re}\left(2\frac{\Delta t^2}{T}\|\mathrm{F}(\lambda - \bar{\lambda})\|^2\right), \tag{B5}$$

where λ is the signal in question, $\bar{\lambda}$ stands for the mean of the signal, $\mathbb{T}$ stands for the fast Fourier transform, and $T = N\Delta t$, with N the number of time iterations with Δt separation.

[1] E. Bairey, I. Arad, and N. H. Lindner, Phys. Rev. Lett. **122**, 020504 (2019).

[2] E. Bairey, C. Guo, D. Poletti, N. H. Lindner, and I. Arad, New J. Phys. **22**, 032001 (2020).

[3] A. Anshu, S. Arunachalam, T. Kuwahara, and M. Soleimanifar, Nat. Phys. **17**, 931 (2021).

[4] J. Haah, R. Kothari, and E. Tang, Optimal learning of quantum Hamiltonians from high-temperature Gibbs states, *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2022), pp. 135–146.

[5] I. Pižorn, V. Eisler, S. Andergassen, and M. Troyer, New J. Phys. **16**, 073007 (2014).

[6] X.-L. Qi and D. Ranard, Quantum **3**, 159 (2019).

[7] A. Zubida, E. Yitzhaki, N. H. Lindner, and E. Bairey, arXiv:2108.08824.

[8] R. Harper, W. Yu, and S. T. Flammia, PRX Quantum **2**, 010322 (2021).

[9] D. S. França, L. A. Markovich, V. V. Dobrovitski, A. H. Werner, and J. Borregaard, arXiv:2205.09567.

[10] A. Gu, L. Cincio, and P. J. Coles, arXiv:2206.15464.

[11] F. Wilde, A. Kshetrimayum, I. Roth, D. Hangleiter, R. Sweke, and J. Eisert, arXiv:2209.14328.

[12] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, Nat. Commun. **5**, 4213 (2014).

[13] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, New J. Phys. **18**, 023023 (2016).

[14] G. Verdon, J. Marks, S. Nanda, S. Leichenauer, and J. Hidary, arXiv:1910.02071.

[15] J. Preskill, Quantum **2**, 79 (2018).

[16] C. Zoufal, A. Lucchi, and S. Woerner, npj Quantum Inf. **5**, 103 (2019).

[17] A. Assouel, A. Jacquier, and A. Kondratyev, Quantum Mach. Intell. **4**, 28 (2022).

[18] C. Bravo-Prieto, J. Baglio, M. Cè, A. Francis, D. M. Grabowska, and S. Carrazza, Quantum **6**, 777 (2022).

[19] S. Y. Chang, E. Agnew, E. F. Combarro, M. Grossi, S. Herbert, and S. Vallecorsa, J. Phys.: Conf. Ser. **2438**, 012062 (2022).

[20] S. Alvi, C. W. Bauer, and B. Nachman, J. High Energ. Phys. 02 (2023) 220.

[21] V. S. Ngairangbam, M. Spannowsky, and M. Takeuchi, Phys. Rev. D **105**, 095004 (2022).

[22] A. Blance and M. Spannowsky, J. High Energy Phys. 02 (2021) 212.

[23] L. Bassman, K. Klymko, D. Liu, N. M. Tubman, and W. A. de Jong, arXiv:2103.09846.

[24] J. Selisko, M. Amsler, T. Hammerschmidt, R. Drautz, and T. Eckl, arXiv:2208.07621.

[25] J. Foldager, A. Pesah, and L. K. Hansen, Sci. Rep. **12**, 3862 (2022).

[26] M. Cerezo, K. Sharma, A. Arrasmith, and P. J. Coles, npj Quantum Inf. **8**, 113 (2022).

[27] S.-X. Zhang, Z.-Q. Wan, C.-K. Lee, C.-Y. Hsieh, S. Zhang, and H. Yao, Phys. Rev. Lett. **128**, 120502 (2022).

[28] M. H. Gordon, M. Cerezo, L. Cincio, and P. J. Coles, PRX Quantum **3**, 030334 (2022).

[29] Y. Du and I. Mordatch, arXiv:1903.08689.

[30] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press, Cambridge, 2012).

[31] A. Huber, in *Mathematical Methods in Solid State and Superfluid Theory*, edited by R. C. Clark and G. H. Derrick (Springer, Boston, 1968), Chap. 14, pp. 364–392.

[32] M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, S. Racanière, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, Phys. Rev. D **106**, 014514 (2022).

[33] K. Cranmer, S. Golkar, and D. Pappadopulo, arXiv:1904.05903.

[34] A. Butter *et al.*, SciPost Phys. **7**, 014 (2019).

[35] G. Kasieczka, T. Plehn, J. Thompson, and M. Russel, https://zenodo.org/records/2603256 (2019).

[36] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, Comput. Phys. Commun. **191**, 159 (2015).

[37] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi (DELPHES 3), J. High Energy Phys. 02 (2014) 57.

[38] M. Cacciari, G. P. Salam, and G. Soyez, J. High Energy Phys. 04 (2008) 063.

[39] M. Cacciari, G. P. Salam, and G. Soyez, Eur. Phys. J. C **72**, 1896 (2012).

[40] J. Y. Araz and M. Spannowsky, J. High Energy Phys. 04 (2021)296.

[41] J. Y. Araz and M. Spannowsky, J. High Energy Phys. 08 (2021) 112.

[42] T. Plehn, G. P. Salam, and M. Spannowsky, Phys. Rev. Lett. **104**, 111801 (2010).

[43] J. Y. Araz and M. Spannowsky, Phys. Rev. A **106**, 062423 (2022).

[44] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, K. McKiernan, J. J. Meyer, Z. Niu, A. Száva, and N. Killoran, arXiv:1811.04968.

[45] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke *et al.*, arXiv:1605.08695.

[46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, *et al.*, tensorflow.org (2015).

[47] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, arXiv:1711.10604.
[48] https://gitlab.com/jackaraz/qhbm.
[49] D. P. Kingma and J. Ba, arXiv:1412.6980.
[50] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, Quantum Sci. Technol. **4**, 024001 (2019).
[51] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Nat. Commun. **12**, 1791 (2021).
[52] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Phys. Rev. A **101**, 032308 (2020).
[53] G. E. Hinton, Neural Comput. **14**, 1771 (2002).
[54] G. E. Hinton, S. Osindero, and Y.-W. Teh, Neural Comput. **18**, 1527 (2006).
[55] G. E. Hinton and R. R. Salakhutdinov, Science **313**, 504 (2006).
[56] J. W. Cooley and J. W. Tukey, Math. Comput. **19**, 297 (1965).