# Physics solutions for machine learning privacy leaks

Alejandro Pozas-Kerstjens,[1,2] Senaida Hernández-Santana,[3] José Ramón Pareja Monturiol,[1,2] Marco Castrillón López,[4] Giannicola Scarpa,[5] Carlos E. González-Guillén,[3] and David Pérez-García[1,2]

[1]*Instituto de Ciencias Matemáticas (CSIC-UAM-UC3M-UCM), 28049 Madrid, Spain*
[2]*Departamento de Análisis Matemático, Universidad Complutense de Madrid, 28040 Madrid, Spain*
[3]*Departamento de Matemática Aplicada a la Ingeniería Industrial, Universidad Politécnica de Madrid, Madrid, 28006, Spain*
[4]*Departamento de Álgebra, Geometría y Topología, Universidad Complutense de Madrid, Madrid, 28040, Spain*
[5]*Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid, 28031, Spain*

Machine learning systems are becoming more and more ubiquitous in increasingly complex areas, including cutting-edge scientific research. The opposite is also true: the interest in better understanding the inner workings of machine learning systems motivates their analysis under the lens of different scientific disciplines. Physics is particularly successful in this, due to its ability to describe complex dynamical systems. While explanations of phenomena in machine learning based on physics are increasingly present, examples of direct application of notions akin to physics in order to improve machine learning systems are more scarce. Here we provide one such application in the problem of developing algorithms that preserve the privacy of the manipulated data, which is especially important in tasks such as the processing of medical records. We develop well-defined conditions to guarantee robustness to specific types of privacy leaks, and rigorously prove that such conditions are satisfied by tensor-network architectures. These are inspired by the efficient representation of quantum many-body systems, and have shown to compete and even surpass traditional machine learning architectures in certain cases. Given the growing expertise in training tensor-network architectures, these results imply that one may not have to be forced to make a choice between accuracy in prediction and ensuring the privacy of the information processed.

Vast amounts of data are routinely processed in machine learning pipelines, every time covering more aspects of our interactions with the world. When the models processing the data are made public, is the safety of the data used for training it guaranteed? This is a question of utmost importance when processing sensitive data such as medical records, but also for businesses whose competitive advantage lies in data quality.

The gold standard in privacy protection within machine learning [1, 2] is provided by differential privacy [3], which consists of inserting carefully crafted noise either in the training dataset [4, 5], in the final model parameters [3], in the objective function [6], or in the gradient updates [7], in order to hide the presence or absence of any particular sample in the training dataset. There exist, however, privacy-related issues that do not directly fall in this category. Imagine a machine-learning algorithm designed to diagnose a specific disease, which uses patients' records as training data, and that these records have a strong imbalance in a particular morbidity which turns out to have no association to the disease target of the model. Even if irrelevant for the final task of the algorithm, knowing this imbalance may have consequences even at the individual level, if the participation of a patient in the study (in contrast with the knowledge of their full record) is disclosed by other means. As a first result, we show that this concern is a reality in machine learning architectures based on neural networks, caused by the driving of the corresponding network parameters to erase the information that is irrelevant.

In regards to the protection of privacy, the ideal model would only retain from the training dataset the information that is essential to perform well. In this sense, access to the model's parameters in one such models would not be more informative about the data used for training than having a description of it by means of recording the outputs generated for different inputs. This is a problem similar in spirit to the protection of software against Man-At-The-End attacks [8]. As a second contribution, we show that the characterization of complex physical systems can provide a fruitful alternative viewpoint on the problem of privacy in machine learning. Concretely, we rigorously prove that in specific tensor network architectures [9], inspired by the theory of efficient representation of quantum many-body states, it is possible and easy to find alternative parametrizations of a model that are as informative as a black-box access to it. Moreover, this is achieved with no impact on the model's performance, in contrast with solutions based on differential privacy.

This work wants to drag focus to physics as a source of mathematically founded inspiration for machine learning solutions. Cross-fertilizations between machine learning and physics are now commonplace [10]: on the one hand, machine learning algorithms are routinely used in particle colliders [11], in the understanding of quantum matter and its properties [12, 13], or in the experimental control of quantum computers [14, 15]; on the other hand, tools developed within the umbrella of physics have proven invaluable in the understanding of the training and performance of machine learning algorithms, perhaps

the most significant being the theory of the information bottleneck [16]. It is expected that such improved understanding leads to new proposals, inspired by physics, of machine learning architectures or training algorithms. However, with the very notable exception of Boltzmann machines [17–19], this type of influence of physics in machine learning has been arguably limited. Our work is one of such form of influences, the main message being that tensor-network architectures provide a favorable framework to develop privacy-preserving machine learning algorithms.

## I. THE VULNERABILITY

Training neural networks is routinely performed via optimization based on gradient descent: given a dataset that wants to be learned and a parametrization of a family of models, a notion of error between the evaluation of the function on the dataset and the expected result is minimized by adjusting the parameters of the model in the direction given by the gradient of the error. Also, most model classes have, at some point in their architecture, a concatenation of parametrized affine transformations, of the form $\boldsymbol{y}^{(l)} = W^{(l)} \cdot \boldsymbol{z}^{(l-1)} + \boldsymbol{b}^{(l)}$ where $W^{(l)}$ is a matrix of weights and $\boldsymbol{b}^{(l)}$ is a vector of biases, and fixed nonlinear functions, $\boldsymbol{z}^{(l)} = \phi_l(\boldsymbol{y}^{(l)})$, applied to the data. These parameters, as we shall see now, contain information about the training dataset that, ideally, a model should not reveal.

For simplicity, let us start considering a hypothetical situation (see Figure 1) where all points in the dataset that we want to learn have one binary feature which takes always the same value, say 1, and the model is a concatenation of these affine transformation and nonlinearities. Since all the datapoints in the dataset have the same value of the binary feature, this information is of no use for the classification, and thus an ideal final model shall not depend on it. During training, the contribution of the biases in the affine transformations are driven to compensate that of the corresponding weights, in the attempt of eliminating the effect of this irrelevant variable on the final prediction. If the initial data has the constant value 1, this will make that the corresponding weights and biases are directed towards taking opposite signs. The situation is the opposite if the binary feature takes the constant value of $-1$. In order to minimize the contribution of the irrelevant feature, in this case the gradient will drive the parameters towards taking values with same signs. An attacker that has access to the model can thus easily recover the nature of the irrelevant feature just by looking at the parity of a set of weights and biases.

In more realistic scenarios the situation is not as clear-cut (after all, if a feature takes the exact same value in the whole dataset, it is not reasonable to feed it to the model). It is more common to have features which have some imbalance between the different values it can take. In such situation, this type of vulnerability is still present, as we demonstrate in Figure 2d. There, we show an illustration with deep neural networks trained on real-world data of medical records derived from the global.health database [21] of COVID-19 cases around the world. The task in which the neural networks are trained is the prediction of the outcome of the infection given demographics, symptoms, and the parity of the date when the case was recorded (this is the feature irrelevant to the task). Then, the attacks follow the spirit of shadow training [22, 23]: the attacker is provided with

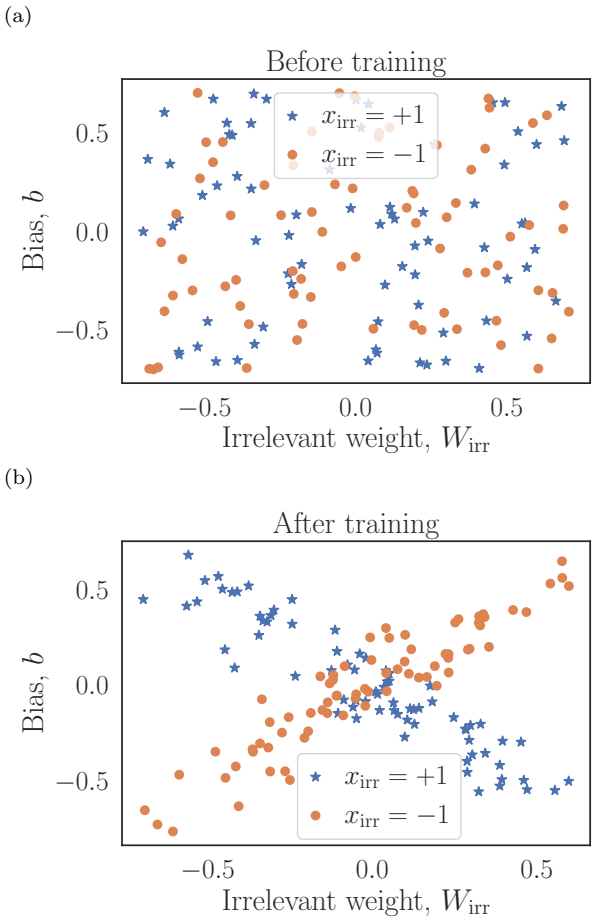



FIG. 1. Illustration of the proposed vulnerability. Each point represents a simple neural network model, $f_\theta(\boldsymbol{x}) = \phi\left(W_{\rm rel}x_{\rm rel} + W_{\rm irr}x_{\rm irr} + b\right)$ (where $\phi$ is an activation function), that is trained to learn the function $y(\boldsymbol{x}) = \text{sign}(x_{\rm rel})$. Each model is trained on a different dataset where $x_{\rm rel} \sim \mathcal{N}(0,1)$, and $x_{\rm irr}$ is $+1$ for all datapoints used to train the models depicted by the blue stars and $-1$ for the orange circles. The plots show the values of the neural network weight for the irrelevant variable, $W_{\rm irr}$, and the bias of the output neuron, $b$, (a) before and (b) after training on a different, random dataset for each model. In this architecture, the gradients of any loss function $\mathcal{L}$ are $\partial_{W_{\rm irr}}\mathcal{L} = x_{\rm irr}\phi'\partial_\phi\mathcal{L}$ and $\partial_b\mathcal{L} = \phi'\partial_\phi\mathcal{L}$, implying that $\partial_{W_{\rm irr}}\mathcal{L} = x_{\rm irr}\partial_b\mathcal{L}$. These gradients will naturally drive the parameters to distinguishable regimes, which can be identified after training as demonstrated in (b). The codes for generating these figures are available in the computational appendix [20].

trained models and labels denoting the majority value of the irrelevant feature in the corresponding training set, and with them trains a meta-model that constitutes the attack. Notably, for the case of neural networks, simple logistic regression meta-models perform well in the attack task, highlighting the vulnerability to the sort of privacy leaks described.

## II. QUANTUM-INSPIRED ARCHITECTURES WITH PRIVACY GUARANTEES

The above example is a (rather extreme) illustration that neural-network architectures store information about the training set in the way that the network parameters process its features. Now we will see that in alternative architectures, inspired by the study of quantum many-body systems, this information can easily be deleted without compromising on model performance.

Continuing with the previous example, it is clear that there exists a straightforward way to erase the information about irrelevant features that does not lead to privacy leaks: simply setting to zero the weights that propagate the influence of those features to the initial layer of the network. In a hypothetical situation where one fixed such parameters and trained the rest, the result would be an alternative collection of weights and biases leading to a model, ideally equally performing, yet not containing any information about the training dataset other than that needed for making the prediction. Thus, the ability to characterize the sets of parameters that lead to the same model opens the door to ways of choosing model parameters which contain no more information about the training dataset than what can be inferred from recording the output for different inputs. The first part of our contribution in this aspect is formally proving this intuition in Section III. The second part is showing, in Section IV, that for the family of matrix product state architectures (this is a family of architectures originally developed in the context of many-body quantum systems that have recently attracted interest in machine learning [24, 25]) one can make an assignment of parameters that has specially suitable properties when analyzed under the lens of privacy in machine learning.

### A. Matrix product states

The field of quantum many-body physics has been developing manageable ways of simulating the states and evolutions of quantum systems composed of many particles, which have recently gained attention in machine learning as alternative parametrizations of high-dimensional, convoluted functions. These are the so-called tensor network architectures and, notably within them, the matrix-product state (MPS) representations [9, 24]. In fact, these parametrizations of complex functions were later rediscovered in the field of numerical analysis, under the name of tensor trains [25, 26]. An MPS architecture is best defined when viewing functions as hyperplanes on high-dimensional feature maps of the input. This is, when we consider (vector) functions $f(\boldsymbol{x})$ of the input $\boldsymbol{x}$ as taking the form

$$f(\boldsymbol{x}) = M \cdot \Psi(\boldsymbol{x}). \tag{1}$$

MPS architectures correspond to the family of functions illustrated in Figure 2b. These are obtained when the vector feature map, $\Psi(\boldsymbol{x})$ (typically non-trainable), has a tensor-product form with one component per dimension of the input, $\Psi(\boldsymbol{x}) = \psi_1(x_1) \otimes \psi_2(x_2) \otimes \cdots \otimes \psi_N(x_N)$, and the hyperplane is expressed as a product of in general complex matrices (hence the name), namely

$$M^{\ell}_{s_1,s_2,\ldots,s_N} = \sum_{\{\alpha\}} A^{\alpha_1}_{s_1} A^{\alpha_1,\alpha_2}_{s_2} \cdots A^{\alpha_j,\alpha_{j+1}}_{\ell} \cdots A^{\alpha_N}_{s_N}. \tag{2}$$

These architectures have a very well-characterized gauge symmetry group, which characterizes the sets of parameters that describe the same function. If between every two consecutive matrices one inserts a decomposition of the identity $\mathbb{1} = Y_j \cdot Y_j^{-1}$ for an invertible matrix $Y_j$, the set of matrices given by $B_{s_j} = Y_j^{-1} \cdot A_{s_j} \cdot Y_{j+1}$ produce $M$ as well. Importantly, it is well-known from quantum many-body physics that these are the only symmetries of the MPS architectures [27], and that it is possible to fix a value of the gauge for each MPS. This fixing is known as a "canonical form", and it is generally obtained via singular value decompositions [27, 28].

## III. PRIVACY FROM REPARAMETRIZATION INVARIANCE

Invariance under reparametrizations is commonly known as gauge symmetry [29]. This is a concept that is commonplace in many-body physics, nuclear and particle physics, or in general relativity. In order to rigorously prove that a complete characterization of gauge symmetries protects against revealing unintended information, we must first define some mathematical objects. For a fixed learning architecture (this is, a functional ansatz depending on $n$ parameters), call $\mathcal{W} \in \mathbb{C}^n$ the set of all possible values of its parameters. The architecture, along with a point $\theta \in \mathcal{W}$, completely determines a model. Thus, we will refer to $\mathcal{W}$ as the set of white-box representations for a given architecture. In this picture, training a model amounts to choosing the optimal $\theta$ for a specific task. In general, this optimal value will depend on the data used for training the algorithm. Analogously, the set of black-box representations $\mathcal{B}$ can be understood as the set of oracular functions (i.e., seen as input-output pairs) corresponding to each $\theta \in \mathcal{W}$. In general, there exists a function $\pi : \mathcal{W} \to \mathcal{B}$ that assigns every white-box representation to its corresponding black-box oracle. In certain cases one can define a right inverse, $\alpha : \mathcal{B} \to \mathcal{W}$,

that assigns a set of parameters to a black-box representation. This function satisfies $\pi \circ \alpha = \mathrm{id}_{\mathcal{B}\to\mathcal{B}}$, this is, that the oracular function associated to a white-box representation of a black box is the black box itself. Due to redundancies in the description, there can be many $\theta \in \mathcal{W}$ that lead to the same black-box representation, so in general, even if an $\alpha$ can be defined, it is not true that $\alpha\circ\pi = \mathrm{id}_{\mathcal{W}\to\mathcal{W}}$. This function, which takes all white boxes describing the same black box to the same element of $\mathcal{W}$, is commonly known as canonical form.

Consider also what an attack is. In the proposed context, an attack is a function applied on a white-box representation whose output provides information about features of the training data. Formally, we can model this as a (smooth) function $f : \mathcal{W} \to \mathbb{C}$ that maps the parameters of a model to a complex number.

With these, we can now state our first result:

**Theorem 1.** *Consider the set of white-box representations for some architecture, $\mathcal{W} \in \mathbb{C}^n$, and its associated set of black-box representations, $\mathcal{B} = \pi(\mathcal{W})$. If a right inverse $\alpha : \pi \circ \alpha = id_{\mathcal{B}\to\mathcal{B}}$ can be defined, then for every function $f$*

$$f(\hat{\theta}) = \hat{f}[\pi(\hat{\theta})],$$

*where $\hat{\theta} = \alpha \circ \pi(\theta) \in \mathcal{W}$ are the parameters denoting a canonical form for the model described by $\theta$ and $\hat{f} = f \circ \alpha$ is the application of $f$ in black boxes.*

*Proof.* By expanding $f(\hat{\theta})$:

$$\begin{aligned} f(\hat{\theta}) &= f \circ \alpha \circ \pi(\theta) \\ &= f \circ \alpha \circ \mathrm{id}_{\mathcal{B}\to\mathcal{B}} \circ \pi(\theta) \\ &= f \circ \alpha \circ \pi \circ \alpha \circ \pi(\theta) \\ &= \hat{f}[\pi(\hat{\theta})], \end{aligned}$$

where we have decomposed the identity in the space of black boxes as $\mathrm{id}_{\mathcal{B}\to\mathcal{B}} = \pi \circ \alpha$. □

The relation above means that the evaluation of an attack, $f$, in a set of parameters that describe the canonical form of a model, $\alpha \circ \pi(\theta)$, coincides with the evaluation of the induced attack, $\hat{f} = f \circ \alpha$, in the associated black box, $\pi(\hat{\theta}) = \pi(\theta)$. Therefore, an attack $f$ cannot extract from a canonical form any information that is not present in the associated black-box.

Note that the only requirement of Theorem 1 is that the function $\alpha$ can be defined. However, in particular, there are no requirements on the degree of regularity, or smoothness, of $\alpha$, so in general the degree of regularity of the black-box attack $\hat{f}$ will depend on the regularity of both $f$ and $\alpha$. In the next section we prove that, for the case of MPS, $\alpha$ can indeed be taken as smooth as possible: namely holomorphic. This implies that the degree of regularity of the white-box and black-box attacks is the same, and therefore, that MPS are strong candidates for privacy-preserving machine learning architectures.

## IV. GLOBAL CANONICAL FORM IN MPS

We now know that if a canonical form can be defined, then the information about the training dataset stored in the model's parameters is the same information that is stored in the corresponding black box. However, it could still be possible that accessing this information is easier when knowing the model's parameters. In this section we prove that this is not the case for MPS architectures. The reason for this is that the map $\alpha$, from the space $\mathcal{B}$ to the space $\mathcal{W}$ described in Section III, can be taken to be holomorphic, and hence possesses the highest degree of smoothness. Then, since holomorphy implies that the composition of any function with $\alpha$ preserves its regularity, any attack at the white-box level, $f : \mathcal{W} \to \mathbb{C}$, can be upgraded to an attack at the black-box level, $\hat{f} = f \circ \alpha : \mathcal{B} \to \mathbb{C}$, with the same regularity. Therefore, for every white-box attack to a canonical-form representation not only there exists a black-box attack with the same performance (as we showed in Theorem 1), but also with the same regularity. This is, not only the canonical form stores as much information as the black box, but also it is equally hard to extract such information in both cases.

In this section we give an explicit construction for such holomorphic map $\alpha$ in MPS architectures. Recall that MPS architectures process the input $\boldsymbol{x}$ via $f(\boldsymbol{x}) = M \cdot \Psi(\boldsymbol{x})$, where $\Psi(\boldsymbol{x})$ is a (typically non-trainable) feature map with a tensor-product form and $M$ was described in Eq. (2) with each entry of the $A$ tensors in the left-hand side being a trainable parameter. Thus, when considering MPS architectures, the set $\mathcal{W}$ is a subset of $\mathbb{C}^n$ (where typically $n = Nb^2d$ for some number of sites $N$, bond dimension $b$, and physical dimension $p$, see Refs. [30, 31] for descriptions of these quantities and their connections with the simulation of quantum many-body systems) given by all the possible values of all the elements of the $A$ tensors in the right-hand side of Eq. (2). The function $\pi$ that maps a white-box representation of an MPS to its corresponding black box is, precisely, Eq. (2). Then, the set $\mathcal{B}$ is defined as $\pi(\mathcal{W})$, i.e., the subset of $\mathbb{C}^{n'}$ (with $n' = d^N$) defined by all allowed values of the entries of the $M$ tensor in the left-hand side of Eq. (2).

In order to provide the map $\alpha$, we will construct a specific holomorphic canonical form for MPS. Formally, this is an holomorphic map $\mathcal{W} \to \mathcal{W}$, transforming each MPS to its canonical form, so that all MPS related by a gauge transformation are mapped to the exactly same canonical-form MPS. This map then induces uniquely the desired $\alpha : \mathcal{B} \to \mathcal{W}$, which will also be holomorphic, as shown below.

The construction of the holomorphic canonical form is done in the following Theorem:

**Theorem 2.** *For an MPS defined by a collection of tensors* $\{A_{s_j}^{\beta_{j-1},\beta_j}\}_{j=1}^N$*, a global, holomorphic canonical form is given by* $\hat{A}_{s_1}^{\beta_1} = \mathbb{1}$*,* $\hat{A}_{s_N}^{\beta_{N-1}} = \mathbb{1}$*, and* $\hat{A}_{s_j}^{\beta_{j-1},\beta_j} = \sum_\gamma L_{s_j}^{\beta_{j-1},\gamma} C_{\gamma,\beta_j}$*, where*

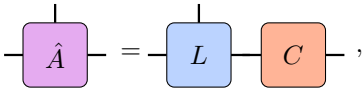
$$L_{s_j}^{\beta_{j-1},\gamma} = \sum_{\{\alpha\}} A_1^{\alpha_1} \cdots A_1^{\alpha_{j-3},\alpha_{j-2}} B_{\beta_{j-1},s_j,\gamma}^{\alpha_{j-2},\alpha_{j+1}} A_1^{\alpha_{j+1},\alpha_{j+2}} \cdots A_1^{\alpha_{N-1}},$$

$$B_{\beta_{j-1},s_j,\gamma}^{\alpha_{j-2},\alpha_{j+1}} = \sum_{\alpha_{j-1},\alpha_j} A_{\beta_{j-1}}^{\alpha_{j-2},\alpha_{j-1}} A_{s_j}^{\alpha_{j-1},\alpha_j} A_\gamma^{\alpha_j,\alpha_{j+1}},$$

$$C = L^{-1}, \text{ with } L_{\gamma,\beta_j} = L_\gamma^{1,\beta_j}.$$

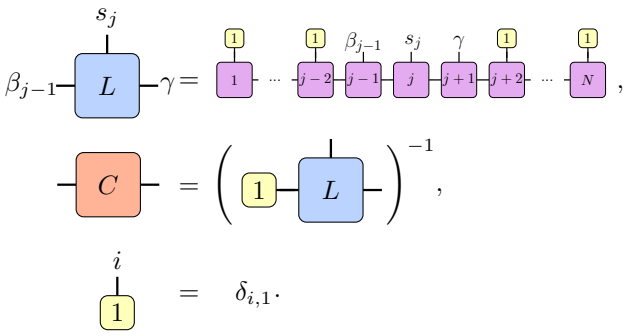
*Proof.* The function is well-defined whenever the $L$ matrices are invertible. In such case, the only non-trivial operation is the computation of such inverses, which is an holomorphic operation in the elements of the matrix. □

Following the usual graphical notation for tensor networks (for its definition see, for instance, the explanations in Figure 2b or Ref. [31]), this decomposition is




(3)

with




Theorem 2 assumes, a priori, that the dimension of all legs are the same. However, it is easy to obtain decompositions with different dimensions, by using projectors different than the one considered. Importantly, note that this decomposition requires that the $L$ matrices are invertible. These matrices are, at every step in the procedure, just a projection on specific sites of the original MPS. While the projected states are not invertible in all MPS, typicality arguments show that this will be the case in general: the set of parameters that give rise to MPS with at least one non-invertible $L$ is of measure zero in the space of all possible parameter values. Note also that the canonical form obtained in this way will be the same for all MPS that are related by a gauge transformation, as required, which means that this canonical form indeed gives a well-defined map $\alpha : \mathcal{B} \to \mathcal{W}$.

It remains to show that $\alpha$ is also holomorphic. In order for this statement to be meaningful, or even to define what it means for an attack on black boxes to be smooth, one needs to endow the set $\mathcal{B}$ with a smooth structure; more specifically, with a complex manifold structure. Note that $\mathcal{W}$ is trivially a complex manifold, since it is an open subset of some $\mathbb{C}^n$.

There are a priori two ways to see $\mathcal{B}$ as a complex manifold. One is as a submanifold, generated by the $\pi$ given by Eq. (2) as $\mathcal{B} = \pi(\mathcal{W})$ and embedded in the ambient (exponentially large) complex vector space of all possible input-output relations. The other is as the space of orbits defined by the gauge symmetries. This is, if we call $\mathrm{S}_{\mathrm{MPS}}$ to the complex Lie group of gauge symmetries, the space of orbits is nothing but the quotient set $\mathcal{W}/\mathrm{S}_{\mathrm{MPS}}$, where a natural complex manifold structure can be defined whenever the symmetry group action is reasonably good (holomorphic, free and proper). One can see, using well-known results in the theory of complex manifolds, that both ways to describe $\mathcal{B}$ are equivalent. Formally, the complex manifolds $\mathcal{W}/\mathrm{S}_{\mathrm{MPS}}$ and $\mathcal{B}$ are biholomorphic, and thus we write $\mathcal{B} = \mathcal{W}/\mathrm{S}_{\mathrm{MPS}}$ from now onwards. The analysis of these manifolds has been done in full detail for the set of "full-rank" MPS in Ref. [30], where we also refer to for the necessary definitions and background. Due to our extra condition of the matrices $L$ being invertible, our sets $\mathcal{W}$ and $\mathcal{B}$ are (full measure) subsets of those considered in Ref. [30], but the exact same proof applies.

By the way the complex manifold structure is defined in the space of orbits $\mathcal{W}/\mathrm{S}_{\mathrm{MPS}}$, for any holomorphic map $\mathcal{W} \to \mathcal{W}$ that is invariant under the action of the gauge group $\mathrm{S}_{\mathrm{MPS}}$ –as it is the case for the canonical form defined in Theorem 2– the uniquely defined associated map $\alpha : \mathcal{B} = \mathcal{W}/\mathrm{S}_{\mathrm{MPS}} \to \mathcal{W}$ is also holomorphic. All details can also be found in Ref. [30] and the references therein.

Summarizing, we have defined a new, holomorphic, canonical form for MPS architectures. This implies that the MPS parameters obtained after computing the canonical form have no more information than the information already included in a black-box oracular access to the model, and that such information is equally hard to extract in both cases. Importantly, the sets of parameters generated by the gauge freedom all describe the exact same function, and thus the reparametrization to a canonical form does not have an impact on the performance of the model. This is in stark contrast with approaches based on differential privacy to protect the privacy in neural networks, where noise is added to the training dataset or the final parameters in a way that

there is a tradeoff between the model's utility and the protection of the dataset.

In particular, for the type of examples discussed in Section I on features that are irrelevant to the target task, and in the ideal scenario in which the model learns perfectly, this means that the MPS parameters obtained after computing the canonical form would have no dependence at all on these irrelevant features. In realistic scenarios such as that illustrated in Figure 2d, one observes that there is still some dependence for high biases, but this is notably lower than that present in the parameters obtained directly from training.

It is important to clarify that our Theorem 2 is just a proof-of-principle illustration of the power of this approach to obtain privacy-preserving machine learning based on tensor networks. We have focused on preserving regularity, but for any other property that one is interested in, in order to obtain a similar reduction from white boxes to black boxes, all that one needs is to find a canonical form that preserves such property. In particular, it is important to note that the canonical form described in Theorem 2 differs from that in the standard literature of MPS algorithms, based on a sequence of singular value decompositions (SVD) [27, 28]. The SVD-based canonical form of MPS presents a residual $\mathrm{U}(1) \times \cdots \times \mathrm{U}(1)$ gauge freedom [30], and it is not clear how to perform a global section of this residual gauge in an holomorphic manner. However, one can a priori expect at least some degree of privacy protection from the SVD-based canonical form. Indeed, relaxing a bit the mechanism, the process of computing the canonical form of an MPS via SVD, and afterwards choosing randomly a representative of the gauge orbit, is equivalent to first computing the canonical form described in Theorem 2, and then computing the SVD-based canonical form of this MPS and randomly choosing a representative of the gauge orbit. The latter process has all the privacy guarantees described in this work because of the application of our canonical form in the first step, and therefore so does the former. Moreover, the analysis on Figure 2 indicates that the direct application of the SVD-based canonical form, without subsequent randomization, also guarantees a significant degree of privacy.

## V. EXPERIMENTS

As an illustration of the protective power of the canonical form of MPS, we depict in Figure 2 the training of one of these MPS architectures in the real-world dataset used in the demonstration with neural networks (the prediction of the outcome of COVID-19 infections given demographics, symptoms, and the parity of the date of the record), as well as the probability of success of attacks based on shadow training [22, 23], both before and after computing its canonical form. This sort of attacks constitute an upper bound to the efficacy of realistic ones because it allows the attacker to have much more information than reasonable. Details on the training and attacks can be found in appendices A and B, respectively. Both types of models (neural networks and MPS architectures) perform very similarly in the target task (see Figure 2c), while the vulnerability to attacks is markedly different: as can be seen in Figure 2d, both neural networks and MPS architectures straight out from training are equally vulnerable but MPS in canonical form are not. For instance, at 80% imbalance of the irrelevant feature, attacks on both models have around 78% of accuracy, while in the canonical-form description of the MPS the accuracy drops down to around 56%, close to the limit of random guessing. Notably, as it is described in Appendix B, this happens despite the attacks performed to MPS architectures being more general than those performed to neural networks.

## VI. DISCUSSION

As machine learning permeates through more layers in society, it is increasingly important to shift the focus from prediction accuracy to greater goals such as privacy and fairness. This work shows that global information about the training dataset is hidden in the parameters of deep learning models, even if this information is irrelevant for the task at hand. More importantly, it points at physics, and more concretely at the tensor networks used in quantum many-body physics, as a favourable framework where to find architectures that are robust to privacy leaks. In this second aspect, our results are encompassed in Theorems 1 and 2, which can be informally stated in the following way:

**Theorem 1** (Informal version)**.** *If the sets of parameters that leave a model invariant can be characterized, then each white-box attack performed on a representative of the set is equally performing (in terms of accuracy) as an attack in the corresponding black box.*

**Theorem 2** (Informal version)**.** *There is a canonical form for the set of MPS architectures so that every white-box attack to such canonical-form set of parameters is "as good" (in terms of the attack accuracy and its regularity as a function) as an attack to the black-box representation.*

We have shown that the key of the protection comes from the ability to characterize all sets of parameters that produce the same trained model. Once this set is characterized, making the final choice of parameters in a way that is independent of the training dataset (i.e., choosing a canonical form) implies that the final parameters contain no more information about the training dataset than the strictly necessary to produce the output. For neural networks, one could imagine obtaining a canonical form by proceeding in the spirit of model extraction attacks [32, 33]. While one would need to study carefully whether the requirements for robustness are satisfied in

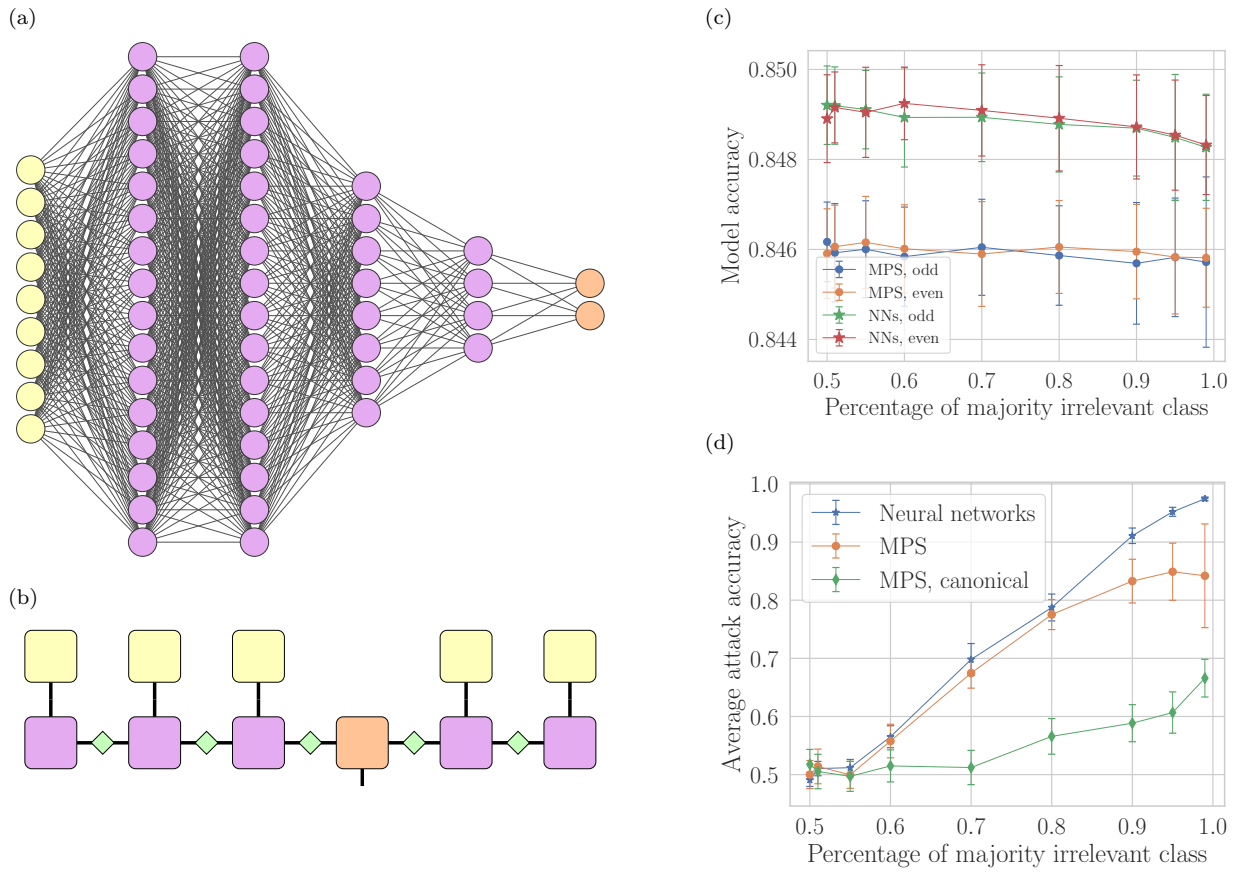



FIG. 2. Comparison between deep neural networks and MPS architectures when learning a model predicting the outcome of COVID-19 infections given demographics and symptoms. Figures of merit are computed as a function of the percentage of dominant value in the irrelevant feature, namely the parity of the day of reporting. For every such percentage, several trainings are run for each of several different datasets, and statistics are computed over the full ensemble of resulting models. Figures (a) and (b) show the neural network and MPS architecture used throughout the experiments. In (b), each element is a tensor with as many dimensions as legs. The purple squares in the lower row represent the parameters of the MPS. They are arranged in three-dimensional tensors, which are multiplied by their neighboring tensors and by the input. This is encoded in the one-dimensional vectors depicted by the yellow squares in the top row. The final tensor after multiplications via Eq. (2) is a vector encoding the output, because the bottom leg of the orange square in the bottom row is free. The gauge symmetry that allows to erase information about irrelevant features is the decomposition of the identity, represented by the green diamonds, in invertible matrices that are later absorbed by the original tensors. Figures (c) and (d) show, respectively, the performance and vulnerability of neural networks and MPS trained on the COVID dataset, as a function of the imbalance between the two values of the irrelevant feature in the training set. The codes for generating these two figures are available in the computational appendix [20]. Figure (c) depicts the average accuracy in the whole database from which the different training sets are generated. The fact that models trained on datasets biased towards different values of the irrelevant feature perform equally indicates that the feature is indeed irrelevant. Figure (d) represents the accuracy of attacks attempting to predict the majority value of the irrelevant feature in the training dataset. Importantly, the MPS not expressed in canonical form are vulnerable in a similar way to the neural networks.

this scenario, one should bear in mind the increased computation time and amount of necessary data (after all, a second model must be trained from input-output queries to the vulnerable one, ideally using datapoints not used in the training of the first), and the potential loss of accuracy entailed by the process.

The standard in privacy protection within machine learning is provided by differential privacy. While now having become the standard in the industry in regards to privacy-preserving machine learning [1, 2], adding noise imposes a tradeoff between the level of privacy and the utility of the model. Moreover, the vulnerability illustrated regards global properties of the dataset rather than individual datapoints. Therefore, whether differential privacy protects against the vulnerability demonstrated remains to be understood.

With this work we want to shift the focus towards a promising class of architectures for machine learning. Our contribution is a foundational step in that direction, where many open questions still lie ahead. We

have provided only an exemplary family of architectures, that of matrix product states, which we prove to admit parametrizations that are no more informative than a black-box access to a trained model. Our results nevertheless apply to any architecture where different parametrizations lead to the same model and where a global, smooth, and one-to-one mapping can be defined between the space of black-box representations and a representative of every possible model. If such mapping exists, one can use it to eliminate information that the final model should not have. This part of the proof is not restricted neither to MPS architectures, to general tensor-network architectures, nor even to neural networks. However, given all the expertise of the community of quantum many-body physics, it is expected that model classes that satisfy the required conditions are easier to find within the tensor-network family [34]. Finding such architecture classes, which are also enough expressive and easy to train, will constitute a very important task.

In contrast with defenses based on differential privacy, the canonical form guarantees that defense in tensor-network architectures can be achieved without compromising on prediction accuracy. However, tensor network architectures can also be trained using differentially private mechanisms, thus adding privacy of individual datapoints on top of the canonical form. Still, more effort must be put into the optimization of training of tensor-network architectures, which only recently are finding advantages over state-of-the-art deep neural network architectures [35]. For this, it is fundamental to develop tensor-network models beyond the highly structured ones developed within the physics community (which, nevertheless, already cover large families of interesting architectures [34]). A good source of inspiration is the "tensorization" of popular deep learning models, a field that is rapidly gaining traction [36–38]. More broadly, both neural-network and tensor-network architectures can easily be combined [39–41], so it is not unreasonable to imagine having hybrid architectures where tensor-network layers take care of some privacy aspects, and neural-network layers do the heavy-lifting on private data.

## ACKNOWLEDGMENTS

We thank Thomas Vidick for insightful discussions and comments. This work is supported by the European Union (Horizon 2020 research and innovation programme-grant agreement No. 648913 and ERDF "A way of making Europe"), the Spanish Ministry of Science and Innovation ("Severo Ochoa Programme for Centres of Excellence in R&D" CEX2019-000904-S and ICMAT Severo Ochoa project SEV-2015-0554, and grants CEX2019-000904-S-20-4, MTM2014-54240-P, MTM2017-88385-P, PGC2018-098321-B-I00 and PID2020-113523GB-I00), and Comunidad de Madrid (PEJ-2021-AI/TIC-23267 and QUITEMAD-CM P2018/TCS-4342).

## Appendix A: Dataset and training of models

To illustrate the fact that neural networks are vulnerable to irrelevant feature leaks and MPS are not, we have trained both architectures in a real-world dataset. This is part of the global.health database of COVID-19 cases [21]. This is a very large database that allows us to make small partitions, so that we can train shadow models when illustrating the attacks. We take the dataset available on March 22nd 2021, use the data of two countries, Argentina and Colombia, for generating our database. The database built only contains the columns `location.country`, `events.outcome.value`, `events.confirmed.date`, `demographics.ageRange.start`, `demographics.gender`, and `symptoms.status` from the whole dataset, and all datapoints where at least one of these entries is empty are discarded. As a first balancing between countries, we take all entries for Argentina and, evenly, one out of every seven (which is approximately the ratio between the number of datapoints for each country) points for Colombia's cases where the column `events.outcome.value` takes the value `Death`, and the same amount for the cases where `events.outcome.value` takes any other value. As a second balancing, now between cases with odd and even registration dates, we take all cases where the column `events.outcome.value` takes the value `Death` (a total of 21 503), and the first 5 375 cases for each combination of country and parity from the subset of points where the column `events.outcome.value` takes the value `Recovered`. We provide the computer codes, written in Python, for generating the database from the global.health dataset in the computational appendix [20].

The classification task in which both, neural networks and MPS, are trained, is in predicting the value for `events.outcome.value` when provided the rest. While all features are discrete in nature, we treat the age feature as continuous. As irrelevant feature, we choose the parity of the report date, extracted from `events.confirmed.date`. While this quantity is indeed expected to be irrelevant for the classification, we check that it is sufficiently uncorrelated with the remaining features (see Table I).

The neural network model used is depicted in Figure 2a and consists of a five-layer architecture with structure 16-16-8-4-2, totalling 614 trainable parameters. Each intermediate layer has a rectified linear unit as activation. Training optimizes the cross-entropy between the predictions and the labels in batches of 8 datapoints, using the Adam optimizer with learning rate of $3\cdot10^{-4}$ and $\ell_2$ regularization of magnitude $6\cdot10^{-3}$, for at most 1 250 epochs. The final model picked is that along the training history that achieves higher accuracy in a held-out validation set composed of 5 000 samples of the original database.

The matrix product state model is depicted in Figure 2b and consist of six tensors where all the dimensions

| | parity | country | age | gender | symptoms | recovery |
|---|---|---|---|---|---|---|
| parity | 1 | 0.005 | 0.001 | 0.001 | −0.012 | −0.002 |
| country | | 1 | 0.059 | 0.033 | 0.161 | −0.055 |
| age | | | 1 | −0.010 | 0.128 | −0.707 |
| gender | | | | 1 | 0.004 | 0.078 |
| symptoms | | | | | 1 | −0.143 |
| recovery | | | | | | 1 |

TABLE I. Pearson correlation coefficients between the columns of the dataset used for training the neural network and MPS models. The columns in the table correspond to the columns (in order) `events.confirmed.date`, `location.country`, `demographics.ageRange.start`, `demographics.gender`, `symptoms.status`, and `events.outcome.value` of the original database.

have cardinality 2. This is, the leftmost and rightmost purple squares in Figure 2b are $2 \times 2$ matrices, and the remaining purple squares and the orange square are tensors of dimensions $2 \times 2 \times 2$. All entries in all tensors (a total of 40) are free, trainable parameters. Training optimizes the cross-entropy between the predictions and the labels in batches of 100 datapoints, using the Adam optimizer with learning rate of $10^{-1}$, for 20 epochs.

The encoding of categorical variables in the input is different for both architectures. For neural networks we perform a traditional one-hot encoding. In contrast, for MPS we perform a noisy encoding, in such a way that for each datapoint the class is encoded in a random value in either $\left[0, \frac{1}{2} - \epsilon\right]$ or $\left[\frac{1}{2} + \epsilon, 1\right]$, with $\epsilon = 5 \cdot 10^{-2}$. Then, every dimension of the input is encoded in a different two-dimensional vector via $\psi(x) = (1 - x, x)$, and these are the objects that are input to the MPS (the yellow squares in Figure 2b). For the `demographics.ageRange.start` column, a min-max normalization is performed before producing the input vector. The different encoding of the variables for neural networks and MPS does not have severe impacts in the vulnerability of the models: as demonstrated in Figure 2d, the MPS architectures are still vulnerable if no protection is applied.

Notably, the function computing the canonical form that is used in the experiments is the standard one based on singular value decompositions. As we commented in Section IV, it is left for future work to analyse whether the SVD-based canonical form fulfills similar regularity properties as the new one we construct here for the proof of Theorem 2. In any case, the analysis on Figure 2 indicates that, in practice, the canonical form obtained via SVD guarantees a significant degree of privacy.

## Appendix B: Attacks

The attacks we consider are in the spirit of shadow modeling attacks [22, 23], whereby the attacker is provided with a large collection of models trained on datasets that share the statistics of the dataset where the victim model has been trained on. This is a situation where the attacker has much more power than what is realistic, so the results are upper bounds to realistic attacks.

In order to produce Figure 2d, we generate, for every percentage of the majority class in the irrelevant variable, a total of 200 datasets (100 for each majority class) with that proportion, randomly sampled from the original dataset. On each of these datasets, a total of 100 models are trained according to the prescription described in 'Dataset and training of models'. This produces, for each of the percentages (the horizontal axis of Figure 2d), 20 000 trained models. Out of all of them, those corresponding to 80 of the datasets are provided to the attacker along with their corresponding majority class, while the models for the remaining 20 datasets will be those to be attacked.

The attacker, depending on the type of models, does a different kind of attack: for neural networks the attack consists of a logistic regressor over the full set of models' weights and biases after a proper normalization, with $\ell_2$ regularization and using the LBFGS solver. This attack is arguably simple, but this only reinforces the argument that neural networks are very vulnerable to leaking the nature of irrelevant features. Removing parameters and keeping those in the initial layer, or doing PCA to reduce dimensionality, did not offer advantages to using the full set of model parameters as input.

For MPS architectures, the attacker trains a deep feed-forward neural network that is input the model weights (after a proper normalization) and outputs the corresponding label. This was done to provide the attack with more generality, so it could hopefully extract more information from the parameters of the model. The attack neural network consists of six layers with structure 20-20-10-10-2-2, using rectified linear units as activation functions for the intermediate layers. Training optimizes the cross-entropy between the predictions and the labels in batches of 1 000 datapoints, using the Adam optimizer with learning rate of $10^{-3}$ and $\ell_2$ regularization of magnitude $10^{-4}$, for at most 1 000 epochs. The set of 80 datasets is split randomly in an 80-20 proportion as to generate a validation set for early stopping.

In order to provide statistics, we train and evaluate the attacks for several random choices of the 80 datasets given to the attacker. We do a total of 1 000 of these repetitions, which provide the confidence intervals in Figure 2.

[1] Apple, Differential privacy overview, https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf (2021), accessed: 2021-12-02.

[2] Google, How we're helping developers with differential privacy, https://developers.googleblog.com/2021/01/how-were-helping-developers-with-differential-privacy.html (2021), accessed: 2021-12-02.

[3] C. Dwork, F. McSherry, K. Nissim, and A. Smith, J. Priv. Confid. **7**, 17 (2017).

[4] S. L. Warner, J. Am. Stat. Assoc. **60**, 63 (1965).

[5] C. Dwork and A. Roth, Found. Trends Theor. Comput. Sci. **9**, 211 (2014).

[6] N. Phan, X. Wu, and D. Dou, Mach. Learn. **106**, 1681 (2017).

[7] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16 (Association for Computing Machinery, New York, NY, USA, 2016) pp. 308–318.

[8] C. Collberg, J. Davidson, R. Giacobazzi, Y. X. Gu, A. Herzberg, and F.-Y. Wang, IEEE Intell. Syst. **26**, 8 (2011).

[9] F. Verstraete, V. Murg, and J. I. Cirac, Adv. Phys. **57**, 143 (2008).

[10] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Rev. Mod. Phys. **91**, 045002 (2019).

[11] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, and T. Wongjirad, Nature **560**, 41 (2018).

[12] J. Carrasquilla, Adv. Phys.: X **5**, 1797528 (2020).

[13] J. F. Rodriguez-Nieva and M. S. Scheurer, Nat. Phys. **15**, 790 (2019).

[14] M. Y. Niu, S. Boixo, V. Smelyanskiy, and H. Neven, npj Quantum Inf. **5**, 33 (2019).

[15] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Phys. Rev. X **8**, 031084 (2018).

[16] N. Tishby, F. C. Pereira, and W. Bialek, The information bottleneck method, arXiv:physics/0004057.

[17] H. C. Nguyen, R. Zecchina, and J. Berg, Adv. Phys. **66**, 197 (2017).

[18] E. W. Tramel, M. Gabrié, A. Manoel, F. Caltagirone, and F. Krzakala, Phys. Rev. X **8**, 041006 (2018).

[19] A. Pozas-Kerstjens, G. Muñoz-Gil, E. Piñol, M. Á. García-March, A. Acín, M. Lewenstein, and P. R. Grzybowski, Mach. Learn.: Sci. Technol. **2**, 025026 (2021).

[20] A. Pozas-Kerstjens and S. Hernández-Santana, Computational appendix of *Physics solutions to machine learning privacy leaks*, GitHub repository (2021), https://www.gihub.com/apozas/private-tn.

[21] Global.health, a data science initiative, https://global.health (2021), accessed: 2021-03-22.

[22] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, Int. J. Secur. Netw. **10**, 137 (2015).

[23] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, in *2017 IEEE Symposium on Security and Privacy (SP)* (2017) pp. 3–18.

[24] E. Stoudenmire and D. J. Schwab, in *Advances in Neural Information Processing Systems*, Vol. 29, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016) pp. 4799–4807.

[25] I. F. Oseledets, SIAM J. Sci. Comput. **33**, 2295 (2011).

[26] I. V. Oseledets, Dokl. Math. **80**, 495 (2009).

[27] D. Pérez-García, F. Verstraete, M. M. Wolf, and J. I. Cirac, Quantum Info. Comput. **7**, 401 (2007).

[28] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).

[29] K. B. Marathe and G. Martucci, *The mathematical foundations of gauge theories* (North Holland, 1992).

[30] J. Haegeman, M. Mariën, T. J. Osborne, and F. Verstraete, J. Math. Phys. **55**, 021902 (2014).

[31] J. I. Cirac, D. Pérez-García, N. Schuch, and F. Verstraete, Rev. Mod. Phys. **93**, 045003 (2021).

[32] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, in *25th USENIX Security Symposium (USENIX Security 16)* (USENIX Association, 2016) pp. 601–618.

[33] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, in *29th USENIX Security Symposium (USENIX Security 20)* (USENIX Association, 2020) pp. 1345–1362.

[34] A. Molnar, J. Garre-Rubio, D. Pérez-García, N. Schuch, and J. I. Cirac, New J. Phys. **20**, 113017 (2018).

[35] J. Wang, C. Roberts, G. Vidal, and S. Leichenauer, Anomaly detection with tensor networks, arXiv:2006.02516.

[36] D. Liu, S.-J. Ran, P. Wittek, C. Peng, R. Blázquez García, G. Su, and M. Lewenstein, New J. Phys. **21**, 073059 (2019).

[37] J. Su, W. Byeon, J. Kossaifi, F. Huang, J. Kautz, and A. Anandkumar, in *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc., 2020) pp. 13714–13726.

[38] X. Ma, P. Zhang, S. Zhang, N. Duan, Y. Hou, D. Song, and M. Zhou, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Curran Associates Inc., Red Hook, NY, USA, 2019) pp. 2232–2242.

[39] I. Glasser, N. Pancotti, and J. I. Cirac, IEEE Access **8**, 68169 (2020).

[40] M. Kuznetsov, D. Polykovskiy, D. P. Vetrov, and A. Zhebrak, in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019) pp. 4102–4112.

[41] S. Cheng, L. Wang, and P. Zhang, Phys. Rev. B **103**, 125117 (2021).